

Open Source Software in CMS

Centers for Medicare and Medicaid Services | Digital Service - Open Source Team

Nicholas Jean, Full-Stack Software Engineer Fellow | UC Berkeley, Computer Science

Brandon Yee, Full-Stack Software Engineer Fellow | Yale, Computer Science

Ricardo Zamora, Full-Stack Software Engineer Fellow | Wesleyan University, Computer Science

Keywords:

LLMs, User Interface, Process Automations, Open-Source (Best Practices, Standards, Metadata), Sorting/Filtering, Python, Automation, Data Visualization, API Endpoints, JavaScript, Full-Stack

Summary:

During their summer internship, Nicholas, Brandon and Ricardo each made significant contributions to various projects. Ricardo laid the groundwork for establishing code.json as an agency-wide metadata standard by adding Cookiecutter prompts to generate code.json files with project metadata. He also worked on both the backend and frontend of the metrics site, creating a new API endpoint for the project code.json fields, adding sorting and filtering features based on project data to improve user experience, and visualizing top committers for each organization. Brandon added several contributions to the repository metrics project, enabling developers to estimate the cost, time, and labor associated with software projects, and to the repository scaffolder project, which automates the input of required open-source documentation. He also added frontend tests to the dedupliFHIR Electron app. Nicholas developed a multimodal LLM assistant via Ollama to help users navigate the office's Murals, created the brand identity for CMS's dedupliFHIR Electron app by designing its icon, added new Cookiecutter CLI variables for the MAINTAINERS.md in the repo-scaffolder project, and used the GitHub API to fetch data on the predominant languages of repositories in the repository metrics project. Together, they laid a strong foundation for these projects and contributed to the agency's open-source initiatives.



DIGITAL SERVICE AT CMS

Coding it Forward Fellowship Final Presentation

Digital Service at CMS // August 2024

Nicholas Jean
Brandon Yee
Ricardo Zamora

Open Source Program Office (OSPO) at CMS

- 1. About CMS and Our OSPO**
- 2. Repo-Scaffolder**
- 3. dedupliFHIR**
- 4. Metrics**
- 5. Open**
- 6. Ollama**



What does the Digital Service at CMS do?

We work to transform the U.S. healthcare system by:



Improving the design of
healthcare experiences



Delivering value to the
government, healthcare
providers, and patients



Modernizing systems



Participating in policy
development

How do we do it?

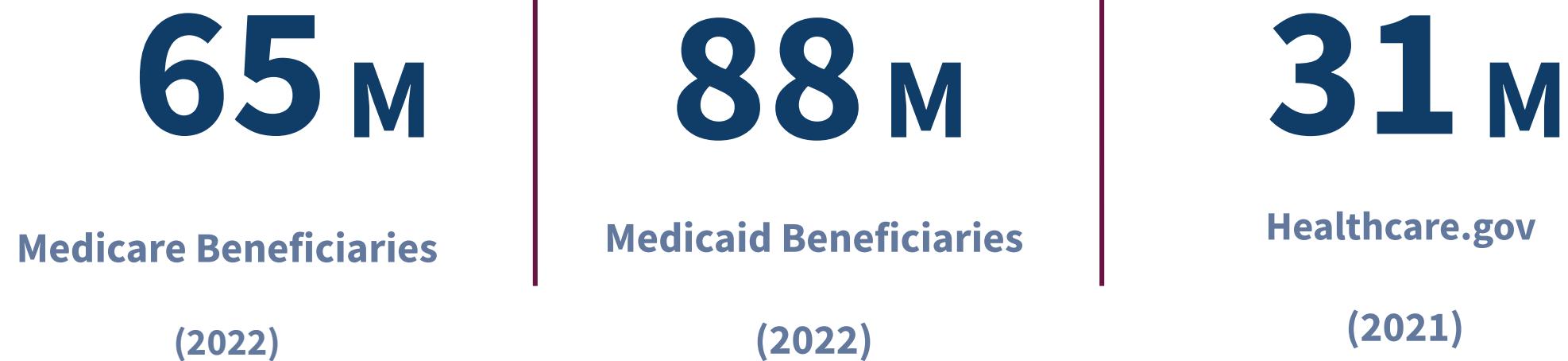
By hiring great talent!

We deploy **small groups** of designers, engineers, and product managers on a "tour of duty" to work alongside **dedicated civil servants**.

These **multidisciplinary teams** bring best practices and new approaches to support government **modernization** efforts.



Who we serve: The American People



<https://data.cms.gov/fact-sheet/cms-fast-facts>

<https://www.cms.gov/files/document/2022-medicare-trustees-report.pdf>



• Who we serve: Taxpayers

\$ 1.7 T

CMS Budget - 12% of the
federal budget

(FY 2022)

\$ 829 B

Total Medicare Payments

(FY 2021)

\$ 646 B

Total Medicaid Payments

(FY 2019)

<https://data.cms.gov/fact-sheet/cms-fast-facts>

<https://www.cms.gov/files/document/2022-medicare-trustees-report.pdf>



• Who we serve: The Health Care System

6,244

CMS Employees

(FY 2022)

1.4M

Health Care Providers

(2022)

20 %

National Health Care
Spending is Medicare

(2022)

<https://data.cms.gov/fact-sheet/cms-fast-facts>

<https://www.cms.gov/files/document/2022-medicare-trustees-report.pdf>



CMS Open Source Program Office (OSPO) Functional Statement:

*“Establishes and maintains guidance, policies, practices, and talent pipelines that **advance equity**, **build trust**, and **amplify impact** across CMS, HHS, and Federal Open Source Ecosystems by working and sharing openly.”*

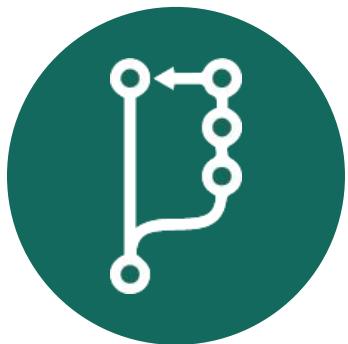


How do we do Open Source at CMS?



Policies

How we **inbound** and **outbound** open source contributions and content



Projects

How we **solve real-world problems** by working in the open



Programs

How we **measure**, and **manage** contributors, projects, **risks**, and **opportunities**

repo-scaffolder

• Purpose

Promote open-source best practices & standards and align with Code.gov's mission by integrating metadata management processes, overall promoting code quality and maintainability.



Tiers

File	Ordinality of Documentation By Tier (M - Mandatory, R - Recommended, N - Not Recommended)				
	Tier 0: Private Repo	Tier 1: One-time Release	Tier 2: Close Collaboration	Tier 3: Working in Public	Tier 4: Community Governance
LICENSE	M	M	M	M	M
SECURITY.md	N	M	M	M	M
README.md	M	M	M	M	M
CONTRIBUTING.md	R	R	M	M	M
MAINTAINERS.md	N	N	R	M	M
GOVERNANCE.md	N	N	N	R	M
CODEOWNERS.md	N	N	R	M	M
COMMUNITY_GUIDELINES.md	N	N	M	M	M
CODE_OF_CONDUCT.md	N	N	M	M	M

Project Description	M	M	M	M	M	This should be 1-3 sentence short description of the project that can be used as a 'one-line' same language as the official 'description' on a GitHub repo landing page.
About the Project	M	M	M	M	M	This should be a longer-form description of the project. It can include history, background, or other supporting materials, or any other information/context that a user or contributor might need to understand the project.
Project Vision	R	N	M	R	R	This should be a forward-looking statement that outlines the desired future state or long-term goals of the project.
Project Mission	R	R	M	R	R	This should be a statement that defines the purpose, scope, and specific objectives of the project.
Agency Mission	R	R	M	R	N	Agency-led projects should include information about their agency mission. This should be a statement that defines the purpose, scope, and specific objectives of the project.
Team Mission	R	R	M	R	N	Agency-led projects should include information about the team executing on the mission. This should be a statement that defines the purpose, scope, and specific objectives of the project.
Core Team	R	M	M	M	M	This information helps with succession planning and provenance for security compliance: understand where the code originated.
Documentation Index	R	R	R	M	M	This is a like a "table of contents" for your documentation. Tier 0/1 projects with simple RE need this, but it is still extremely helpful to provide "bookmark" or "anchor" links to specific other communication channels.
Repository Structure	R	R	R	M	M	Using the "tree" command can be a helpful way to generate this information, but, be sure to use the --dirs option.
Development and Software Delivery Lifecycle	N	R	R	M	M	Tier 1: Even if the lifecycle is "one-time release" being explicit is better than implicit.
Local Development	R	R	M	M	M	Use step by step instructions to get from 'zero' to 'running code.' Should include any system requirements or dependencies needed to run the project. When possible, including install instructions for multiple operating systems the project was developed on is a recommended practice.
Coding Style and Linters	R	R	M	M	M	This section outlines best practices contributors should follow to reduce friction and improve the project. Often times, these checks can be automated and run as part of a continuous integration pipeline.
Branching Model	R	R	R	M	M	Even for Tier 0/1 projects with one or a few contributors, branching models (such as git flow) feature development history clear, and to help reinforce development best practices.
Contributing	N	R	M	M	M	For projects that accept contributions, point towards the CONTRIBUTING.md file. For the section, instead, mention one-time release, or private repo status instead.
Codeowners	N	R	M	M	M	Though all tiers have an 'implied' code-owner, since there is at least one author of the repository may outline the employment or contract of the original author, a shared inbox or an open source email address (e.g. opensource@cms.hhs.gov)
Community	R	R	M	M	M	For projects that are higher tier than a one-time release, pointing your contributors toward online discussion boards or channels, project backlogs and documentation, etc...)
Community Guidelines	R	R	M	M	M	Project tiers above one-time release should point towards a CODE_OF_CONDUCT.md file for conduct and reporting mechanisms and escalation strategies. It is better to have these focus on support if/when there is an incident. (e.g. Contributor-covenant.org)
Governance	N	N	R	R	M	Make a short statement about how the project is governed (formally, or informally) and link to the governance document.
Feedback	N	R	R	M	M	Direct users towards the channel where they're encouraged to provide feedback, typically a forum or a dedicated feedback page.
Glossary	R	R	R	R	M	Even for early-tier projects, this documentation can be extremely valuable. Good candidates for Government projects) and any critical Subject Matter Expertise related vocabularies the project is within.
Policies	M	M	M	M	M	This section is to explicitly link to Federal policies and guidelines that are required or recommended by law. Examples include HIPAA, FERPA, FOIA, and other regulations.
Public Domain	M	M	M	M	M	A best practice is to list the LICENSE under which a project is released at the bottom of the file, default to Creative Commons Zero 1.0 International (world-wide public domain).



• Automation of Contributors

Contributors

Total number of contributors: 7

 Natalia Luzuriaga	 April	 Steve Van Tuyl	 decause-gov	 Isaac Milarsky	 satwic007
 Sebastián					

• Slugification of MAINTAINERS.md

Tier 4

Maintainers

This is a list of maintainers for this project. See [CODEOWNERS.md](#) for list of reviewers for different parts of the codebase. Team members include:

Maintainers:

- @arthur
- @dutch
- @hosea

Approvers:

- @john
- @abigail
- @jack

Reviewers:

- @uncle
- @charles
- @sean

Roles	Responsibilities	Requirements	Defined by
member	active contributor in the community	multiple contributions to the project.	PROJECT GitHub org Committer Team
reviewer	review contributions from other members	history of review and authorship in a subproject	MAINTAINERS file reviewer entry, and GitHub Org Triage Team
approver	approve accepting contributions	highly experienced and active reviewer + contributor to a subproject	MAINTAINERS file approver entry and GitHub Triage Team
lead	set direction and priorities for a subproject	demonstrated responsibility and excellent technical judgement for the subproject	MAINTAINERS file owner entry and GitHub Org Admin Team

Contributors

Total number of contributors:

Tier 2

Maintainers

This is a list of maintainers for this project. See [CODEOWNERS.md](#) for list of reviewers for different parts of the codebase. Team members include:

Maintainers:

{list or table including the fields: role, name, affiliation, github username}

Role	Name	Github Username	Affiliation
reviewer	tony stark	tstark	dsac
approver	pepper potts	ppotts	ccsq

Contributors

Total number of contributors:



• Slugification of CODEOWNERS.md

Code Owners

[@decause-gov](#) [@natalialuzuriaga](#) [@IsaacMilarky](#)

Shoutouts

[@usdigitalresponse](#)

- [@aprilselby88](#)
- [@vantuyls](#)
- [@sgarciahelguera](#)

Thank you [US Digital Response](#) Team for your support creating this repository!

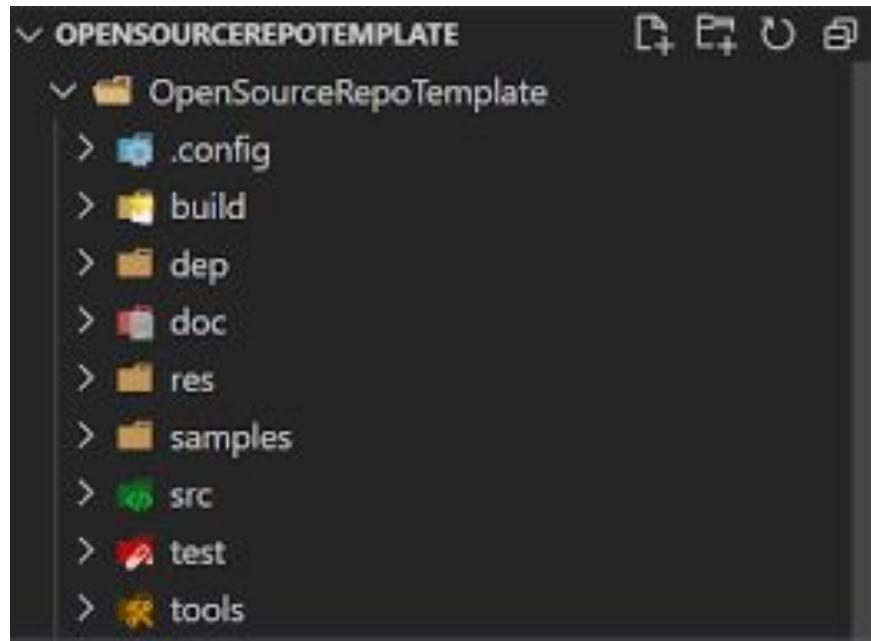
Questions?

Email opensource@cms.hhs.gov



• Generate Repository Structure Documentation

tree -d



Code.json

A standardized metadata file used by U.S. federal agencies to catalog their open-source projects.

```
{  
  "version": "2.0",  
  "agency": "HHS",  
  "measurementType": {  
    "method": "projects"  
  },  
  "releases": [  
    {  
      "name": "pillbox_docs",  
      "description": "Pillbox at the National Library of Medicine",  
      "status": "Development",  
      "permissions": {  
        "usageType": "openSource",  
        "licenses": null  
      },  
      "organization": "Department of Health and Human Services",  
      "homepageURL": "https://github.com/HHS/pillbox_docs",  
      "downloadURL": "https://api.github.com/repos/HHS/pillbox_docs/downloads",  
      "repositoryURL": "https://github.com/HHS/pillbox_docs.git",  
      "vcs": "git",  
      "labourhours": 0.0,  
      "languages": [  
        "Ruby"  
      ],  
      "date": {  
        "created": "2009-12-12",  
        "lastModified": "2021-11-08",  
        "metadataLastUpdated": "2022-03-22"  
      },  
      "tags": [  
        "Pillbox",  
        "National",  
        "Library",  
        "Medicine"  
      ],  
      "contact": {  
        "email": "HHS_GitHub_Service_Desk@hhs.gov",  
        "name": "U.S. Department of Health & Human Services"  
      },  
      "name": "healthdata.gov",  
      "description": "No description available...",  
      "status": "Development",  
      "permissions": {  
        "usageType": "openSource",  
        "licenses": null  
      },  
      "organization": "Department of Health and Human Services",  
      "homepageURL": "https://github.com/HHS/healthdata.gov",  
      "downloadURL": "https://api.github.com/repos/HHS/healthdata.gov/downloads",  
      "repositoryURL": "https://github.com/HHS/healthdata.gov.git",  
      "vcs": "git",  
      "labourhours": 0.0,  
      "languages": [  
        "JavaScript",  
        "XSLT",  
        "Java",  
        "Awk",  
        "CSS",  
        "Shell"  
      ],  
      "date": {  
        "created": "2017-03-19"  
      }  
    }  
  ]  
}
```

HHS code.json

publiccode.yml

International standardized metadata file particularly used by Europe to document and share information about open-source software projects.

Examples

The publiccode.yml Standard

A metadata description standard for public software that is easy to use both for developers and people with less technical background in order to make the software developed by Public Administrations and Public Organisations easily discoverable.

[release v0.4.0](#) [Slack channel](#) [#publiccode](#) [slack](#) [get invite](#)

[Charter](#) | [Voting procedure](#) | [Contributing](#)

What publiccode.yml is for

Many great software projects are developed by public administrations, however reuse of these projects is very limited. Some of the reasons for low uptake of such projects is a lack of discoverability and that it is hard to find out what project can actually work in the context of a different public administration.

The publiccode.yml file is meant to solve all those problems. It is an easily readable file for civil servants that are trying to figure out whether a project will work for them, and easily readable by computers as well. It contains information such as:

- the title and description of the project in English and/or other languages;
- the development status, e.g. concept, development, beta, stable, obsolete;
- which organisation developed the project;
- who is caring for the maintenance and when this expires;
- who to contact for technical or support inquiries;
- what national and local legal frameworks the project is designed for;
- what software dependencies the project has.

The publiccode.yml file format should both be able to easily be added to any new project, as well as grow with the project as it expands beyond the original context it was developed in.

Finding projects

```
rer.bandl / publiccode.yml ⓘ  
starsara Update publiccode.yml ⓘ  
1 # This repository adheres to the publiccode.yml standard by including this  
2 # metadata file that makes public software easily discoverable.  
3 # More info at https://github.com/italia/publiccode.yml  
4  
5 publiccodeVersion: "0.2"  
6 categories:  
7 - application-development  
8 - civic-management  
9 logo: docs/rer-logo.png  
10 dependency:  
11 - open  
12 - names Place  
13 - optional: false  
14 - version: ''  
15 - versionMax: ''  
16 - versionMin: '4'  
17 description:  
18 -!  
19 features:  
20 - Bandi  
21 - Ricerca  
22 - Tile  
23 - Portlet  
24 genericName: Plone add-on  
25 longDescription: >  
26 - Plugin per Plone che aggiunge un nuovo tipo di contenuto "Bandi" con un tipo di cartella speciale per gli allegati, con due diverse visualizzazioni di dettaglio, una vis  
27 - Questo contenuto è folderish (come le cartelle), e permette di impostare informazioni tipiche dei bandi come le scadenze per la partecipazione e la data di chiusura pro  
28 - Inoltre nelle sue cartelle di approfondimento si possono inserire link e allegati.  
29  
30 - Nei risultati di ricerca, come nelle viste di tile e portlet e del contenuto, sono sempre visibili le informazioni sullo stato del bando (aperto, in corso, chiuso) e le  
31 - shortDescription: Modulo per la pubblicizzazione dei bandi per finanziamenti e opportunità  
32 - screenshots:  
33 - - docs/Bandi_dettaglio_vista-a-destra.jpg  
34 - - docs/Bandi_dettaglio_vista-a-sinistra.jpg  
35 - - docs/Bandi_motore-di-ricerca.JPG  
36 - - docs/Bandi_motore-di-ricerca.JPG  
37 - - docs/Bandi_motore-di-ricerca.JPG  
38 - - docs/Bandi_motore-di-ricerca.JPG  
39 developmentStatus: stable
```



• Adding code.json

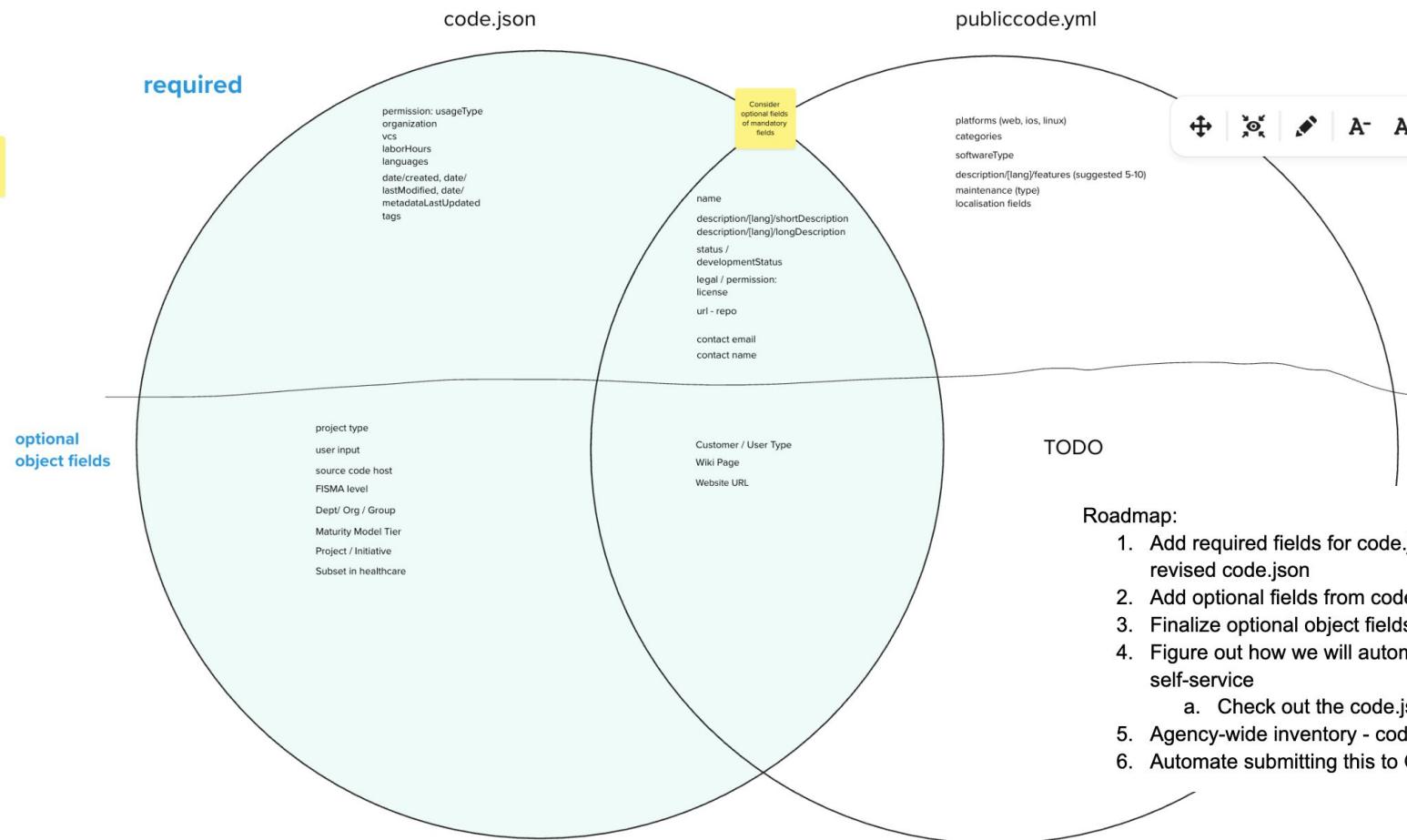
First Implementation:

```
1  {
2      "project_type" : ["Package", "Website", "Standards", "Libraries", "Data", "Apps", "Tools", "APIs"],
3      "user_input": ["Yes", "No"],
4      "project_fisma_level": ["Low", "Moderate", "High"],
5      "group": "CMS/OA/DSAC",
6      "subset_in_healthcare": "Policy, Operational",
7      "user_type": "Providers, Patients, Government",
8      "repository_host": ["Github.com", "GitHub ENT", "GitHub Cloud", "GitLab.com", "GitLab ENT", "GitLab ENT CCSQ"],
9      "__prompts__": {
10          "group": "Which group is the project part of?",
11          "subset_in_healthcare": "Which subset of healthcare does the project belong to?",
12          "user_type": "Who are the intended users?",
13          "user_input": "Does the project accept user input? (e.g. allows user to query a database, allows login by users, etc.)",
14          "repository_host": "Where is the repository hosted?"
15      }
16 }
```



• Adding code.json

What's Next?



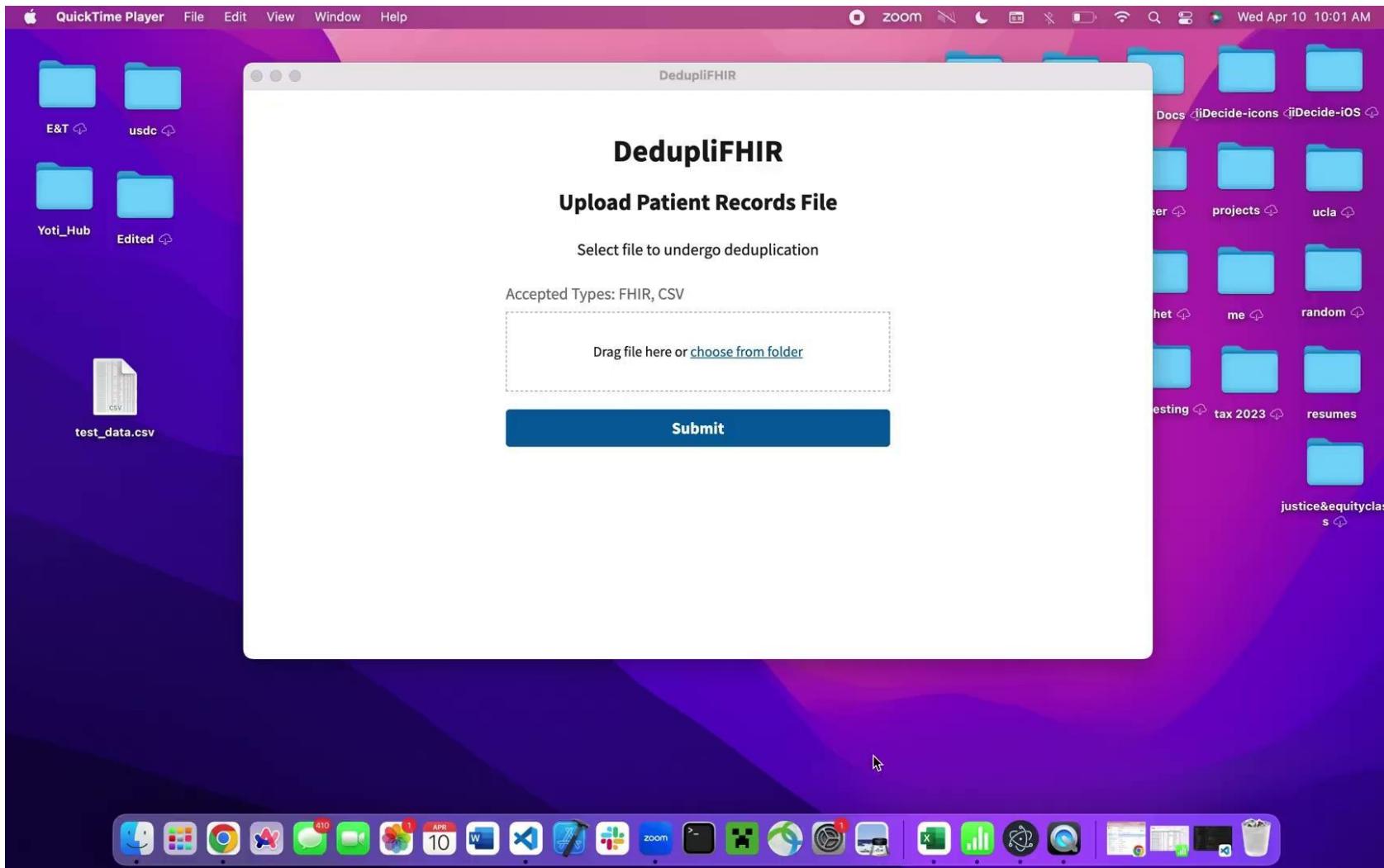
deduplifHIR

• Purpose

dedupliFHIR de-duplicates and aggregates eCQM (*electronic Clinical Quality Measure*) data to ensure accurate and comprehensive quality reporting in healthcare

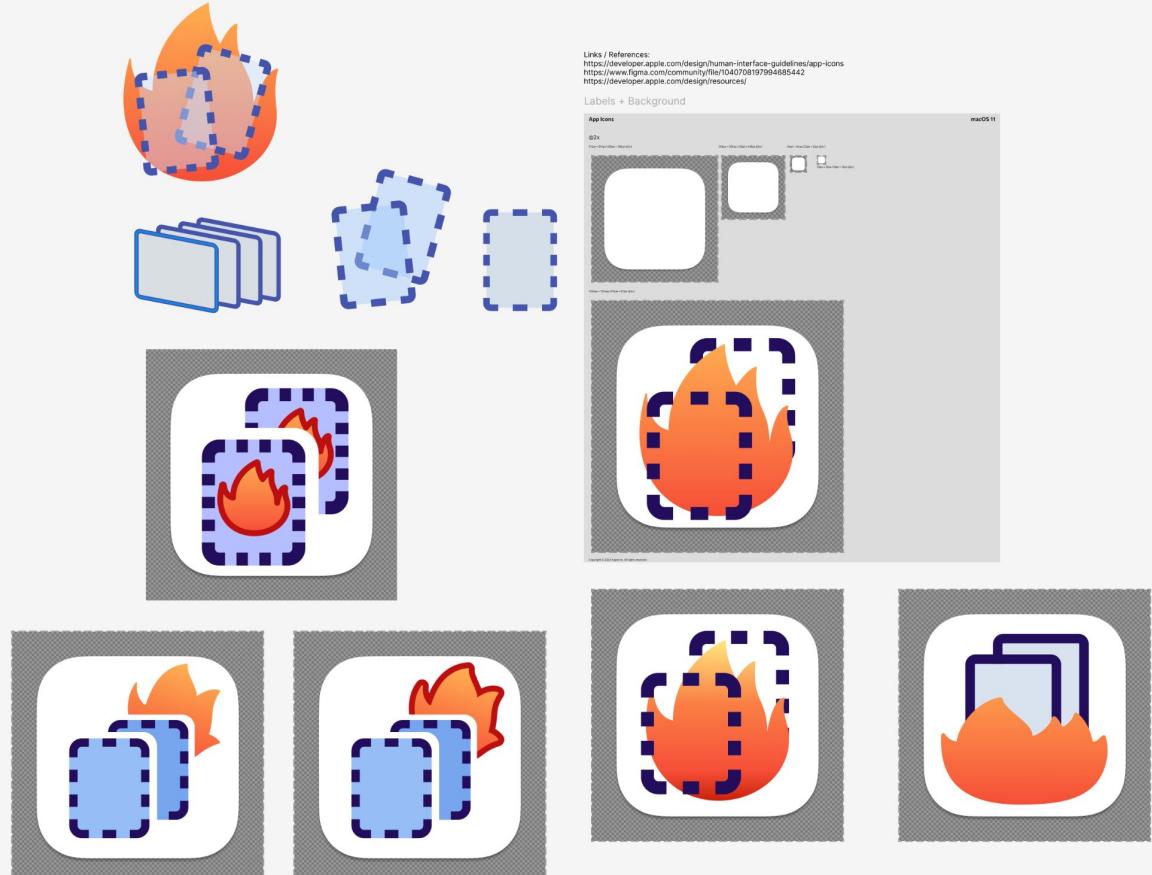


• Demo Video

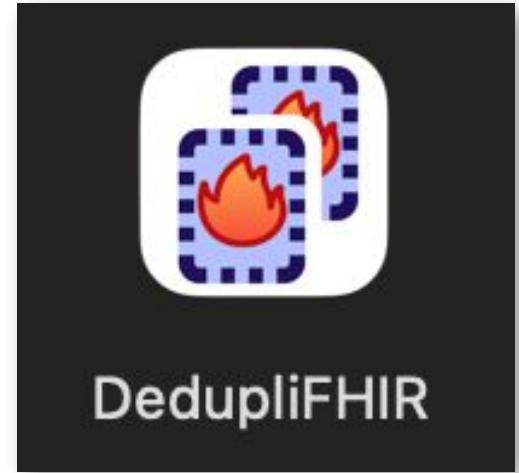


• Electron App Icon

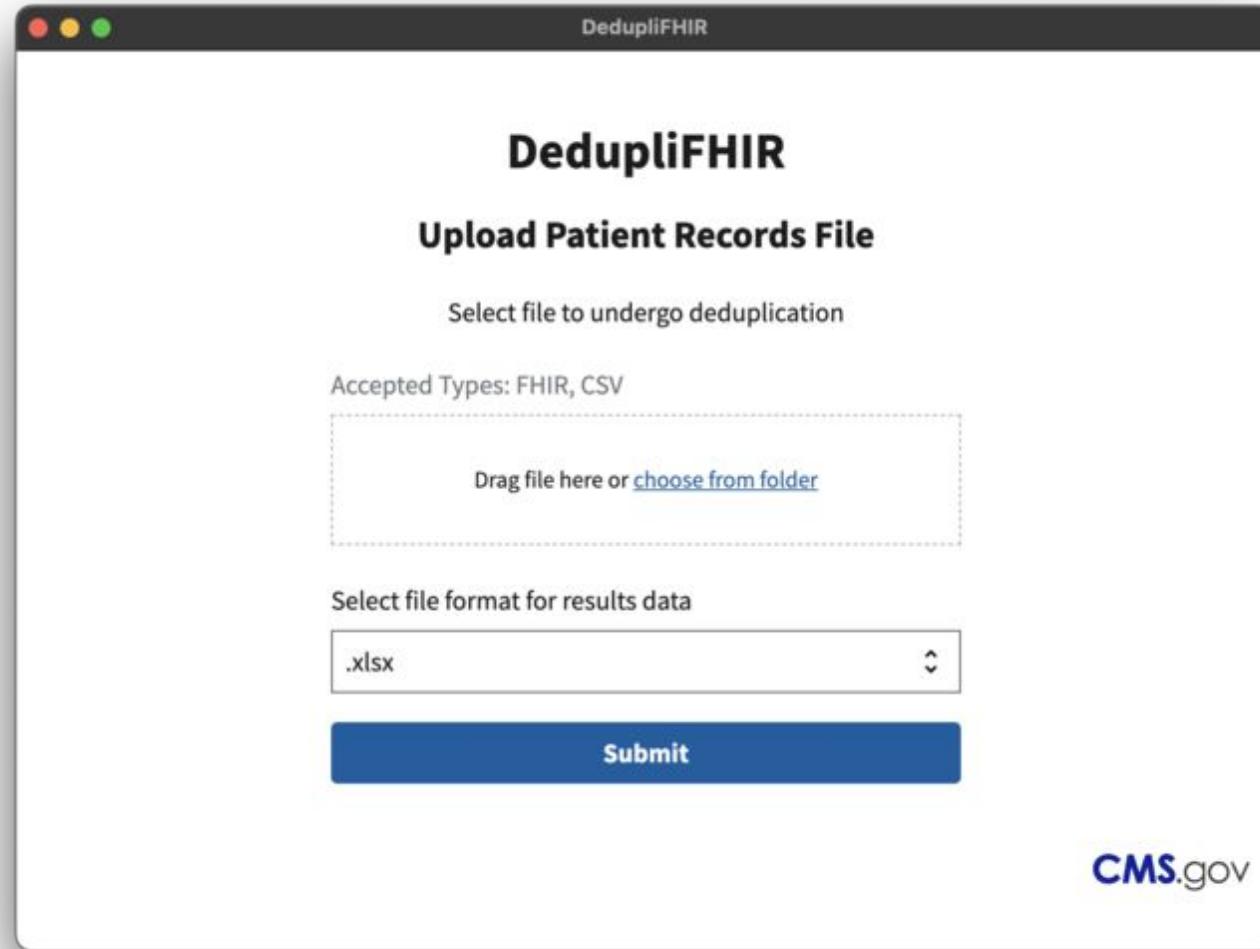
Design



End Result



Frontend Tests



Metrics

• Purpose

Metrics provide valuable insights into various aspects of open source projects to foster a more effective and transparent development environment.



Metrics Website

The screenshot shows a web browser displaying the CMS.gov Open Source Repository Metrics website. The URL in the address bar is dsacms.github.io/metrics/. The page has a light blue header bar with the text "Work in progress. We welcome questions and suggestions — [give us feedback.](#)". Below this, the main content area features the CMS logo and the title "Metrics Website". A navigation bar at the top right includes links for "Home", "Organizations", and "Projects", with "Home" being the active tab. Two main sections are displayed: "Organizations" (with a link to "View Metrics on our GitHub Organizations") and "Projects" (with a link to "View Metrics on our 164+ Repositories"). The footer is dark with white text, featuring the CMS logo, the URL open.cms.gov, and the text "An official website of the Department of Health and Human Services and the Centers for Medicare and Medicaid Services". It also includes links for "About CMS", "FOIA requests", "Office of the Inspector General", "Privacy policy", "Accessibility statement", "No FEAR Act data", and "Performance reports". A note at the bottom says "Looking for U.S. government information and services? [Visit USA.gov](#)".



• Predominant Languages

List repository languages

Lists languages for the specified repository. The value shown for each language is the number of bytes of code written in that language.

Fine-grained access tokens for "List repository languages"

This endpoint works with the following fine-grained token types:

- [GitHub App user access tokens](#)
- [GitHub App installation access tokens](#)
- [Fine-grained personal access tokens](#)

The fine-grained token must have the following permission set:

- "Metadata" repository permissions (read)

This endpoint can be used without authentication or the aforementioned permissions if only public resources are requested.

Parameters for "List repository languages"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Code samples for "List repository languages"

Request example

GET /repos/{owner}/{repo}/languages

cURL JavaScript GitHub CLI

```
curl -L \
  -H "Accept: application/vnd.github+json" \
  -H "Authorization: Bearer <YOUR-TOKEN>" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  https://api.github.com/repos/OWNER/REPO/languages
```

Response

Example response

Response schema

Status: 200

```
{  
  "C": 78769,  
  "Python": 7769  
}
```

Example:

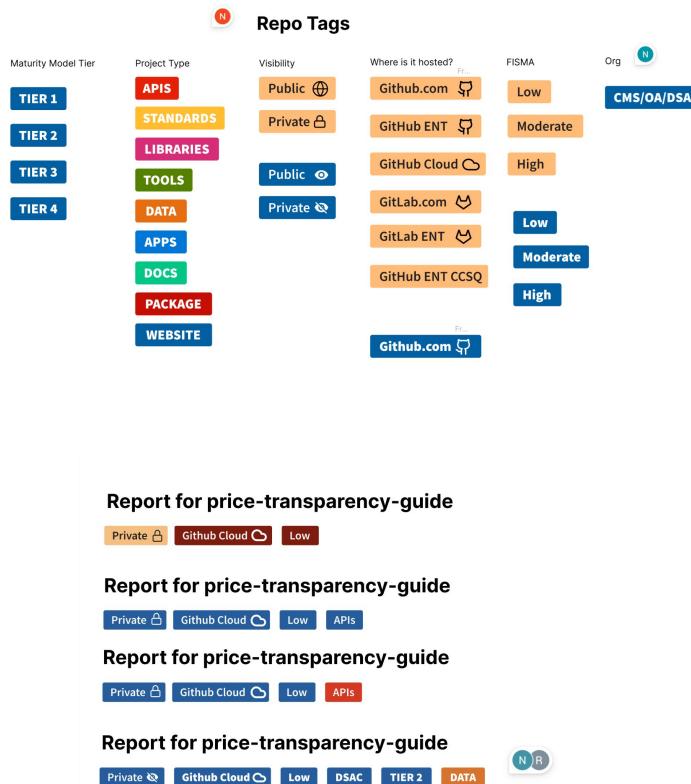
```
{  
  "languages": {  
    "Python": 100000,  
    "JavaScript": 50000,  
    "HTML": 30000  
  }  
}
```

```
# Predominant Languages Endpoint (ex. https://api.github.com/repos/chaoss/augur/languages)  
LANGUAGE_ENDPOINT = "https://api.github.com/repos/{owner}/{repo}/languages"
```



• Repo Metrics Tags

Design



End Result

**CMS.gov Open Source
Repository Metrics**

Report for dedupliFHIR

GITHUB.COM **CMS/OA/DSAC**

TOOLS **TIER 3** **MODERATE**



• Projects Page: New Look!

Design

The screenshot shows the CMS Open Source Repo Metrics interface. At the top, a banner reads "Work in progress. Not an official government site." Below it, a sub-banner states "An official website of the United States government [Here's how you know](#)". The main navigation includes "Home", "Monthly Metrics", and "Weekly Metrics". The "Home" tab is selected. On the left, there are several filter panels:

- Organization:** CMS-Enterprise, CMSgov, DSACMS, Enterprise-CMS.
- Maturity Model Tier:** Tier 1, Tier 3.
- Fisma/Security Level:** Moderate.
- Project Type:** Tools, Website.
- Visibility:** (empty)

The central area is titled "Projects" and lists six items, each represented by a small thumbnail and a summary card:

Project Name	Description	Star Rating	Forks	Issues	Open PRs
price-transparency-guide	The technical implementation guide for the tri-departmental price transparency rule.	★ 307	12	49	8
price-transparency-guide	The technical implementation guide for the tri-departmental price transparency rule.	★ 307	12	49	8
price-transparency-guide	The technical implementation guide for the tri-departmental price transparency rule.	★ 307	12	49	8
price-transparency-guide	The technical implementation guide for the tri-departmental price transparency rule.	★ 307	12	49	8
price-transparency-guide	The technical implementation guide for the tri-departmental price transparency rule.	★ 307	12	49	8
price-transparency-guide	The technical implementation guide for the tri-departmental price transparency rule.	★ 307	12	49	8

On the right side, there are "Sort By" and "Sort Direction" dropdown menus. The "Sort By" menu includes options like Fisma/Security Level, Project Type, Stars, Maturity Model Tier, Project Name, Organization, and Forks. The "Sort Direction" menu has "Ascending" and "Descending" options.



• Projects Page: New Look!

End Result

Search Sort by

Organization

- DSACMS
- CMS-Enterprise
- CMSgov
- Enterprise-CMCS

DSACMS 😊

dedupliFHIR
Prototype for basic deduplication and aggregation of eCQM data

☆ 8 ⚡ 0 ○ 21 📈 78 ⚡ 6

Maturity Model Tier

- Tier 1
- Tier 2
- Tier 3
- Tier 4

metrics
Experimentations in Open Source Repository Metrics

☆ 6 ⚡ 2 ○ 34 📈 156 ⚡ 6

open

☆ 2 ⚡ 1 ○ 29 📈 6 ⚡ 5

repo-scaffolder
Templates and commandline tools for creating repositories for US Federal open source projects

☆ 21 ⚡ 7 ○ 32 📈 90 ⚡ 3

End Result (Mobile)

Organization

Maturity Model Tier

FISMA Compliance Level

Project Type

Search Sort by Sort Direction

- DSACMS
- Low
- Moderate

DSACMS 😊

dedupliFHIR
Prototype for basic deduplication and aggregation of eCQM data

☆ 8 ⚡ 0 ○ 21 📈 78 ⚡ 6



• Projects Page Demo (Video)

The screenshot shows a web browser window with the following details:

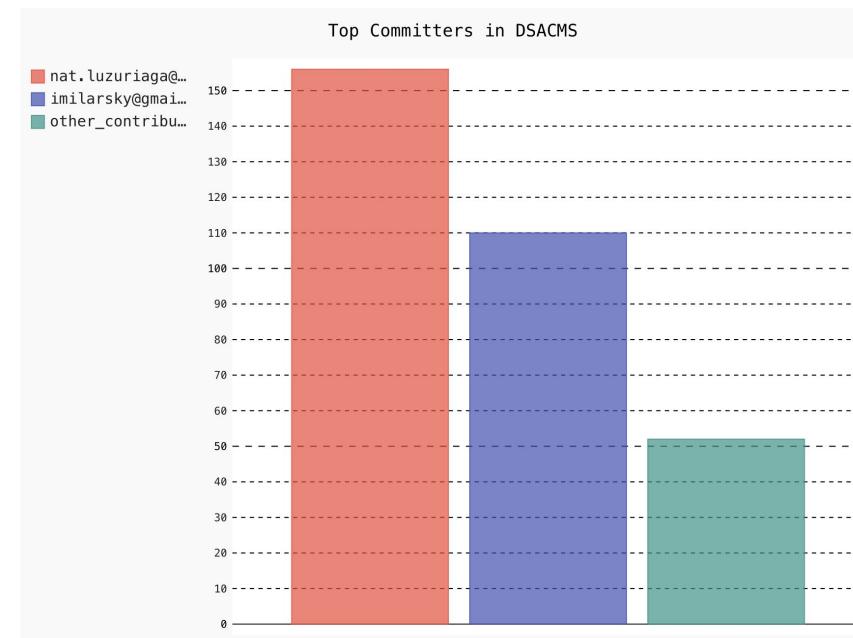
- Address Bar:** Not Secure 0.0.0.0:8080
- Header:** Work in progress. We welcome questions and suggestions — [give us feedback.](#)
- Page Title:** CMS.gov Open Source Repository Metrics
- Navigation:** Home, Organizations, **Projects**
- Section:** CMS Metrics Website
- Text:** Powered by the Open Source Program Office (OSPO) at the Digital Service at the Centers for Medicare and Medicaid Services.
- Buttons:**
 - Organizations** >> View Metrics on our GitHub Organizations
 - Projects** >> View Metrics on our 164+ Repositories
- Footer:**
 - open.cms.gov
 - An official website of the [Department of Health and Human Services](#) and the [Centers for Medicare and Medicaid Services](#)
 - [About CMS](#), [FOIA requests](#), [Office of the Inspector General](#), [Privacy policy](#)
 - [Accessibility statement](#), [No FEAR Act data](#), [Performance reports](#)
 - Looking for U.S. government information and services? [Visit USA.gov](#)



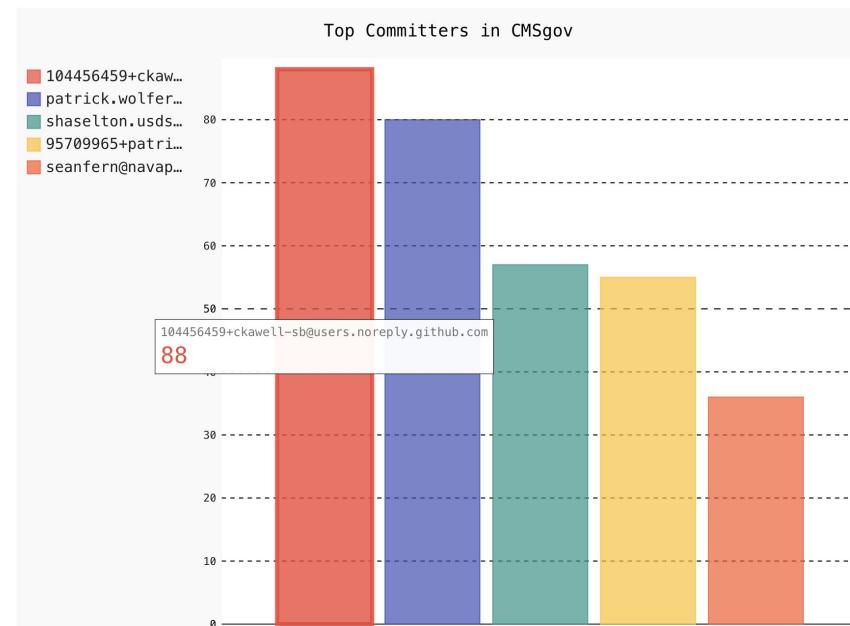
• New Metric: Top Committers by Org

Recognizing developers who have made significant contributions to the project.

Top Committers



Top Committers



open.cms.gov

An official website of the Department of Health and Human Services and the Centers for Medicare and Medicaid Services

About CMS

FOIA requests

Office of the Inspector General

Privacy policy

Accessibility statement

No FEAR Act data

Performance reports



open.cms.gov

An official website of the Department of Health and Human Services and the Centers for Medicare and Medicaid Services

About CMS

FOIA requests

Office of the Inspector General

Privacy policy

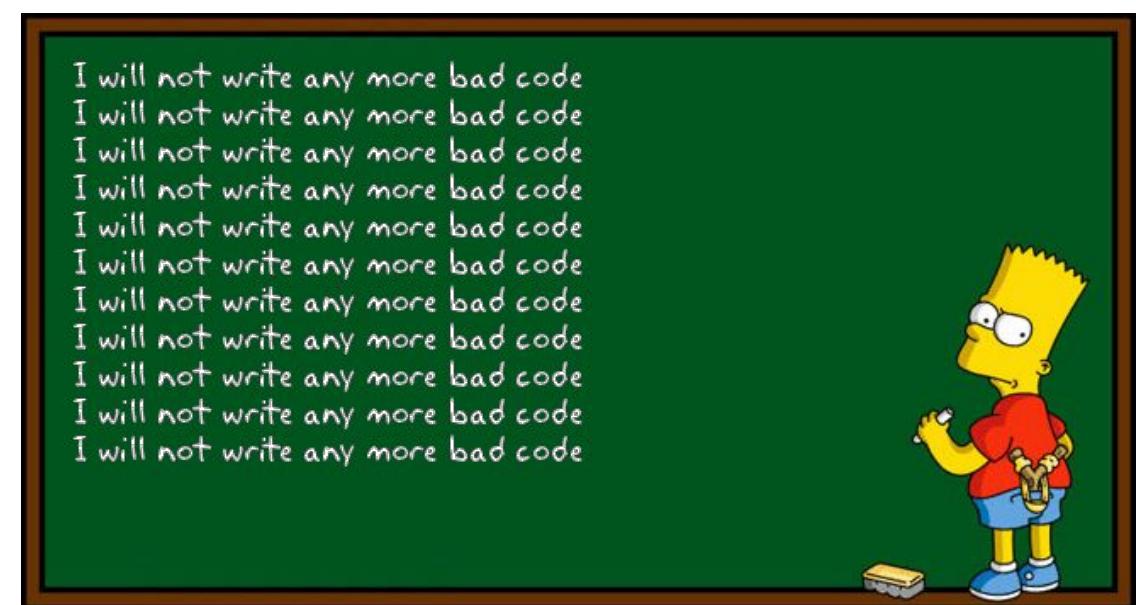
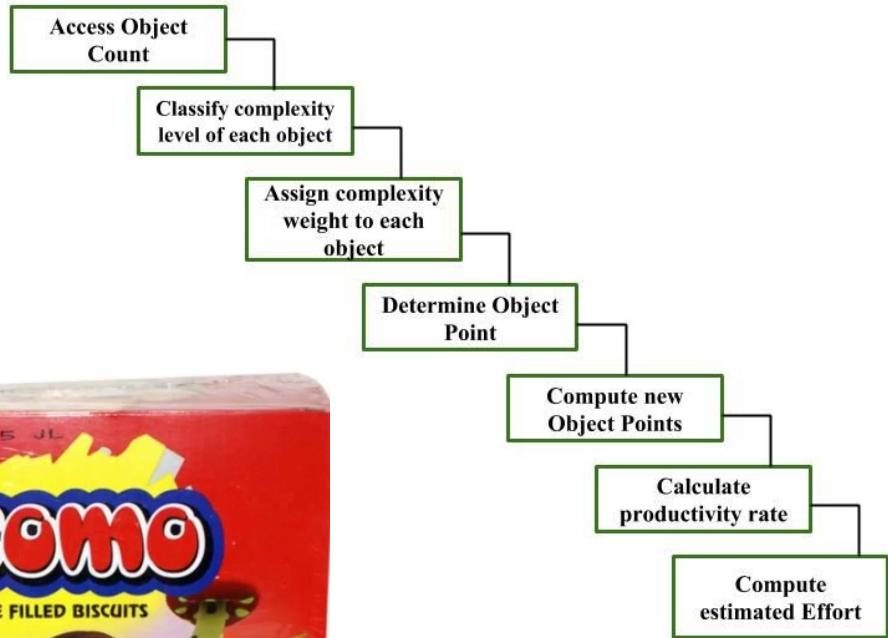
Accessibility statement

No FEAR Act data

Performance reports



• COCOMO & DRYness Scores



• OpenSSF Internal Scorecards

✓ 10	Dangerous-Workflow	CRITICAL
✗ 0	Signed-Releases	HIGH
✗ 0	Token-Permissions	HIGH
✓ 9	Binary-Artifacts	HIGH
✓ 10	Code-Review	HIGH
✓ 10	Maintained	HIGH
✓ 10	Vulnerabilities	HIGH
✗ 0	Fuzzing	MEDIUM
✗ 0	Pinned-Dependencies	MEDIUM
? 4	SAST	MEDIUM
✓ 10	Security-Policy	MEDIUM
✗ 0	CII-Best-Practices	LOW
✓ 10	License	LOW
?	Branch-Protection	HIGH
?	Packaging	MEDIUM



open

• Purpose

Public-facing website with list of CMS open-source projects, precursor to Metrics project

dsacms.github.io/open/projects



Bug Fixes

The screenshot shows the CMS Open Source page. At the top, there's an orange banner with the text "Work in progress. Not an official government site." Below it, a smaller banner says "An official website of the United States government" with a link "Here's how you know". A "MENU" button is visible. The main content area has two sections: "fqm-execution-service" and "cqm-validators", each with a link to "projecttacoma/[service]". Below these is a search bar labeled "Filter". On the left, there's a sidebar with "price-transparency-guide" (STANDARDS) and "design-system" (LIBRARIES). The "design-system" section includes links to "CMSgov/design-system" and "design.cms.gov". At the bottom, there's a footer with links to "USA.gov" and "Visit USA.gov".

← Before

The screenshot shows the CMS Open Source page after bug fixes. The top banner now correctly displays "An official website of the United States government" and "Here's how you know". The "MENU" button is present. The main content area has been simplified, showing only the "Open Source at CMS" header and a "Filter" search bar. The "fqm-execution-service" and "cqm-validators" sections have been removed. The sidebar on the left now contains only the "design-system" (LIBRARIES) section, which has been updated to reflect the changes. The CMS logo and footer information remain at the bottom.

After →



MurAI - Mural LLM Assistant

**Prototype*

• Purpose

A multimodal LLM assistant via Ollama for helping users better understand and navigate our office's Murals



• Starting Models / Platforms

*LLaVa 🌋



LLaVa-Llama3



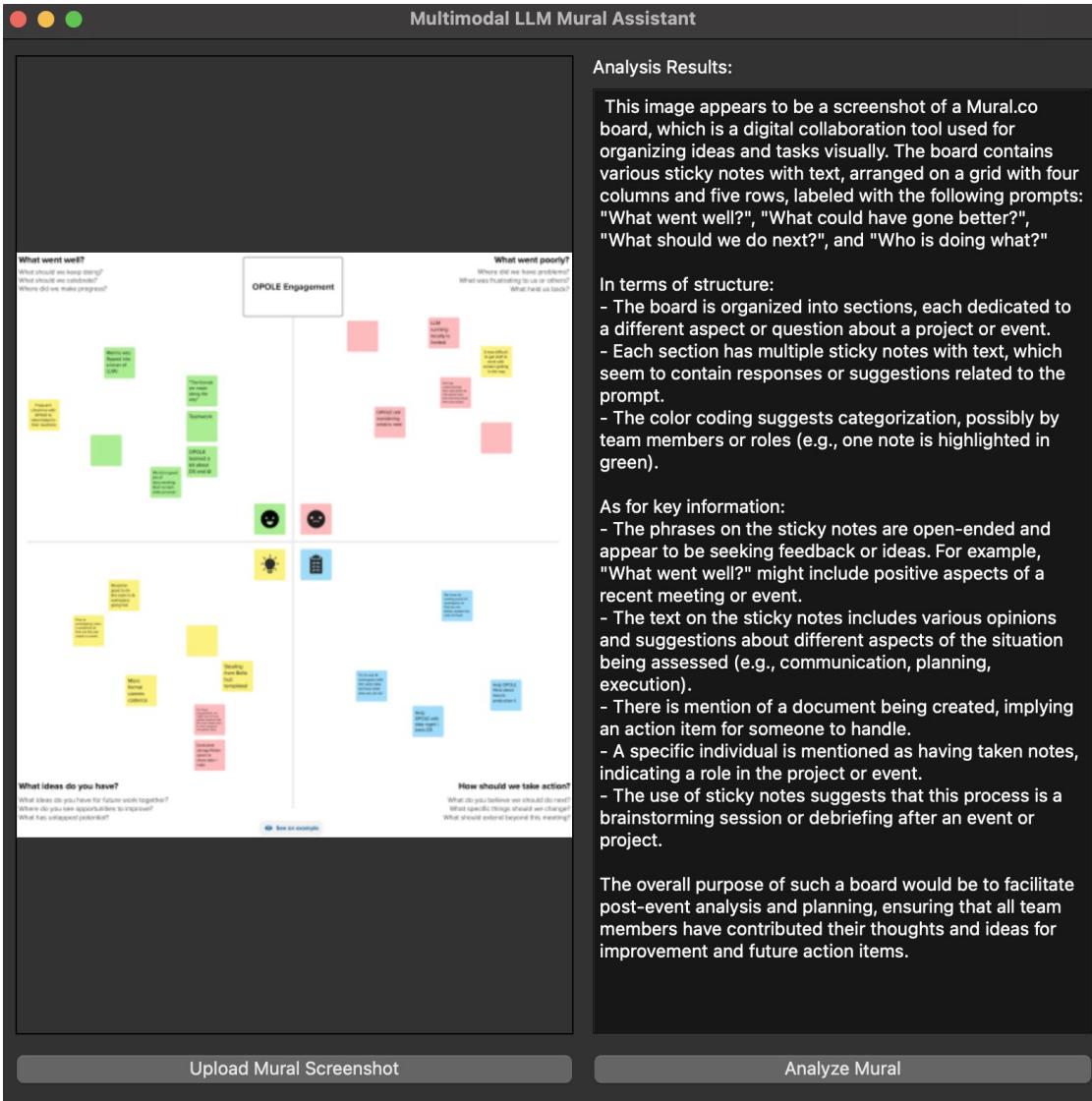
Llama 3.1 🐾

Mixtral

Ollama (platform)



User Interface



First iteration of GUI

- Bare bone functionality
 - Can only analyze Mural screenshot once
 - Cannot chat with LLM or continue asking questions
- Uploaded image has low preview resolution



User Interface

Mural Assistant

The screenshot shows a Mural.co project titled "OPOLE Engagement". The board is organized into several sections:

- What went well?** (Top left): Contains sticky notes like "Marina was exposed into a lover of LMs.", "Frequent users OPOLE is shown lots of their workbooks.", and "We did a good job of documenting our career development".
- OPOLE Engagement** (Central top): A large section containing "The friends we made along the way!", "Teamwork", "OPOLE learned a lot about DS and AI", and "OPOLE still wondering what it's next".
- What went poorly?** (Top right): Contains notes like "LLM running locality is limited", "It was difficult to get off to start, never getting in the way", and "Not fully understanding what went wrong, then preconceived".
- What ideas do you have?** (Bottom left): Contains notes like "What to do to make this work in AI better and going forward", "More team comms cadence", "Stealing from Bechtel but templates", "Help OPOLE with data insight / basic OR", and "Help OPOLE think about how to prioritize it".
- How should we take action?** (Bottom center): Contains notes like "We have to understand what went wrong and learn from it to prevent it from happening again", "Dedicated independent space to work on ML code", and "Ask a question...".
- Chat:** (Right side): A text-based analysis of the mural.

You: Analyze this mural

MurAI: The image shows a screenshot of a Mural.co project, which is an online collaborative whiteboard tool. Mural.co allows users to create visual boards with sticky notes and organize them into different sections for group brainstorming and collaboration purposes.

The board in the screenshot has several sections and sticky notes with text on them. The main sections are:

1. "What we did?" – This section seems to be a summary or recap of activities that took place during a project or meeting. There is one sticky note here, which reads "What we did?", indicating a space for team members to write down their contributions or tasks completed.
2. "What we went well?" – This section is likely used to capture positive aspects of the work done together. It contains two sticky notes: "What we went well?" and "We didn't go wrong". These suggest that the team reflects on what worked well and might also consider any mistakes made during the project or meeting.
3. "What we can learn from?" – This section seems to be a space for reflection, where the group identifies areas for learning and improvement. There is one sticky note here with a question, "What we can learn from?", inviting team members to share insights on what could be learned from the project or meeting outcomes.
4. "How should we do?" – This section might be used to discuss future strategies or methods for improving efficiency and effectiveness in projects or meetings. There are two sticky notes here: "What should we do?" and "We didn't do". These could suggest areas of focus for improvement or aspects that need attention in the next round of work.
5. "How do we know it's good?" – This section is intended to provide a space to evaluate outcomes and determine if they meet desired standards or goals. It contains three sticky notes: "How do we know it's good?", "We don't know", and "We know". These could be responses from team members on how to assess the quality of their work, whether they have clear indicators of success, or if there are specific criteria for what constitutes a successful outcome.

The colors of the sticky notes vary, with some in yellow, blue, green, and pink, providing visual differentiation between sections and ideas. The overall structure suggests a focus on reflection, learning from experience, and planning future improvements. Key information appears to be related to team collaboration, project evaluation, and professional development.

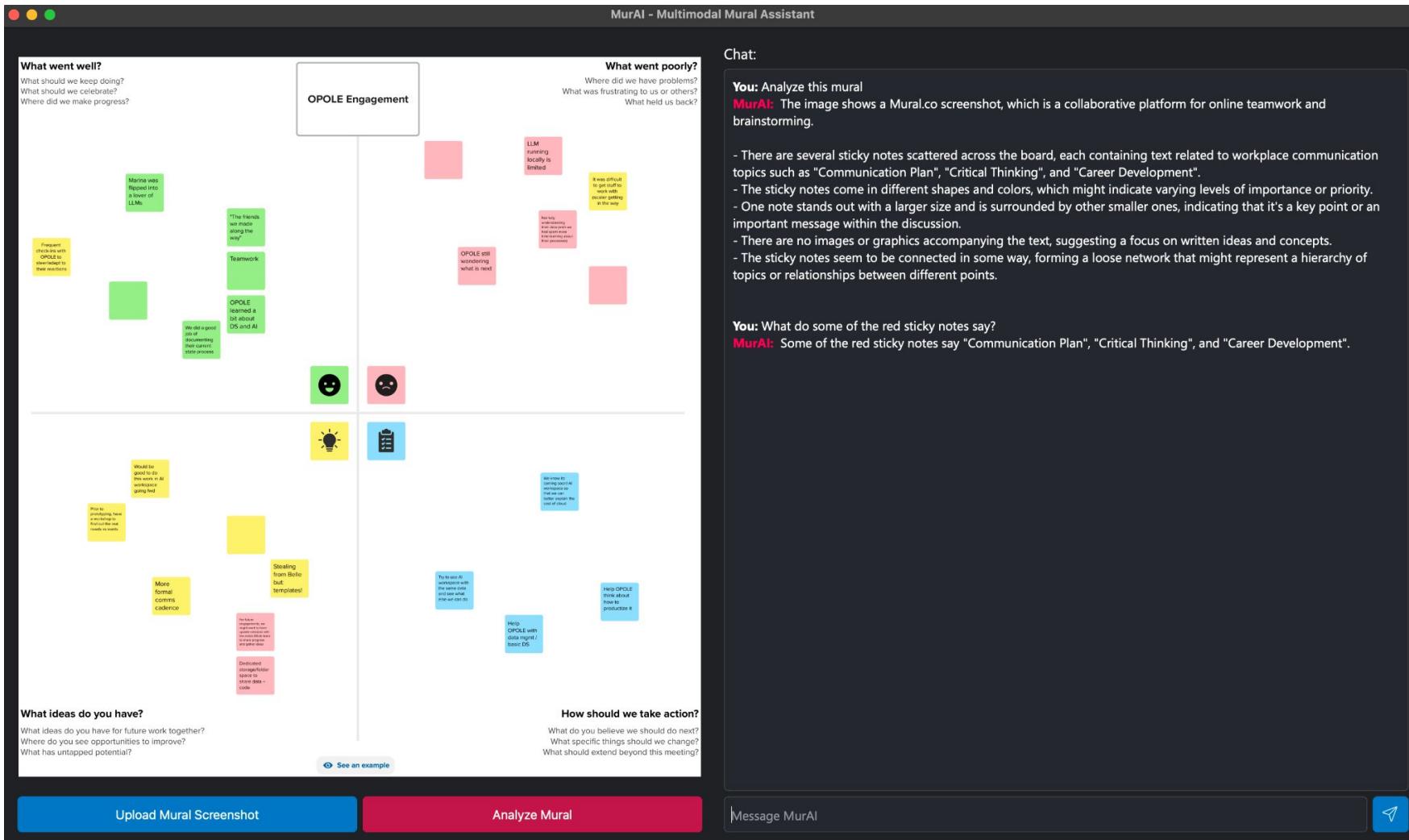
Upload Mural Screenshot **Analyze Mural**

Several iterations later...

- Now able to ask follow-up questions
- GUI retouch
- Now called “MurAI”!
- Uploaded image preview now at full resolution



User Interface

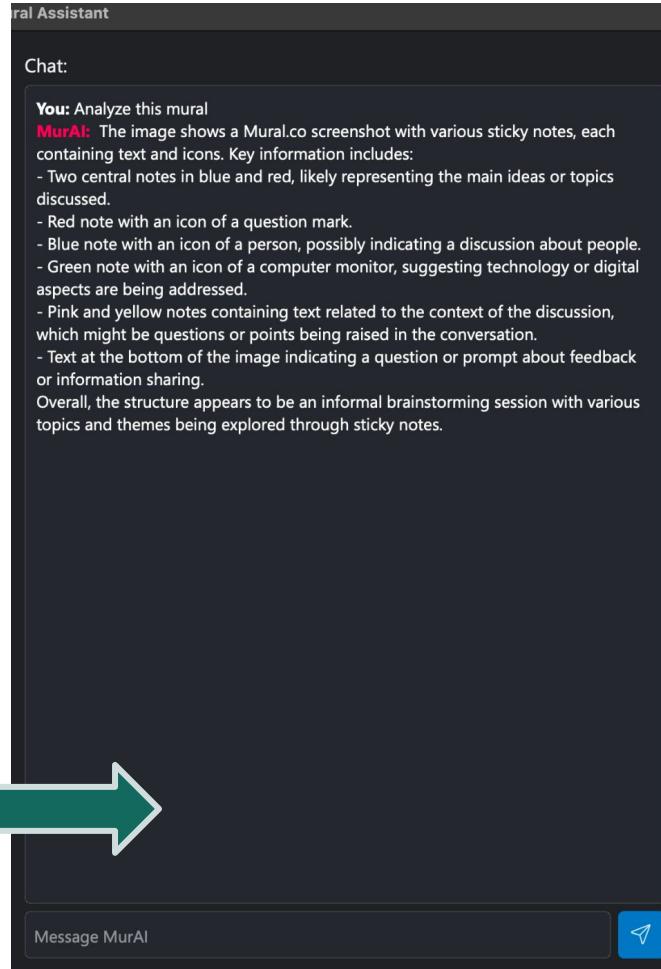
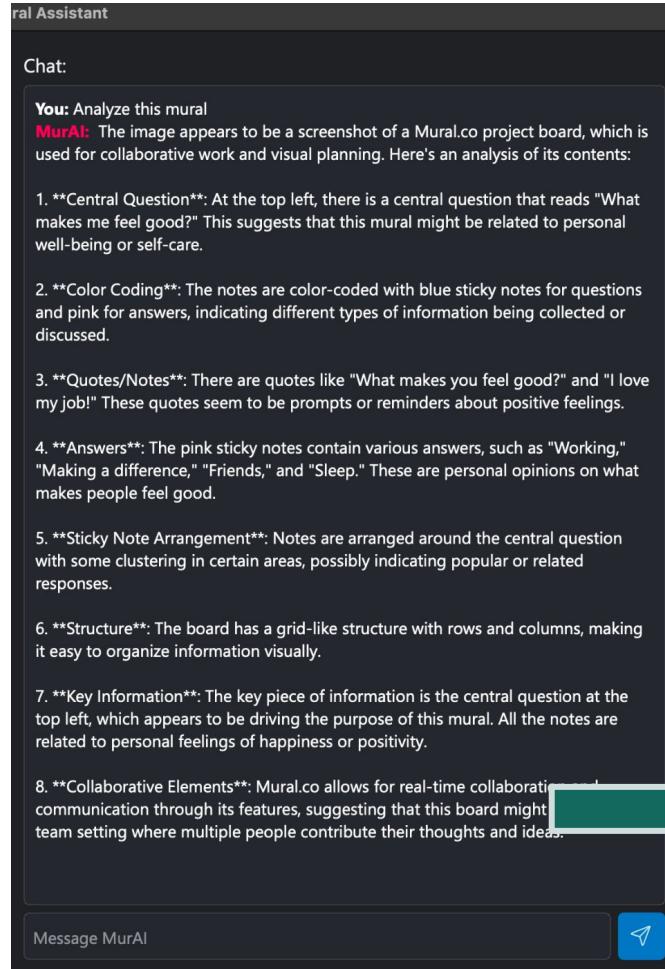


Final Iteration

- Moved “Analyze Mural” button to left side
- Added send message button (right side)
- Color correction!



• LLM Fine-tuning



Goal: “Analyze Mural” should be brief but detailed enough for the user to have an overall understanding of the Mural

- Be as specific as possible!
- Give the LLM a role
- List format for instructions
- Tipping culture
- Order matters

• The Future and Next Steps

- Other methods of fine-tuning
 - Breaking down the screenshot into smaller chunks/tokens
 - Training the model with Mural screenshots
 - OCR (Optical Character Recognition) tool integration
 - Confidence scores
- Better models
 - LLaVa:13b (**8 GB**) vs LLaVa:34b (**20 GB**)
- Better GPUs
 - AWS cloud GPUs for greater computational power



Thank You DSAC!

- **Thank you!**

- DSAC team
- Remy DeCausemaker
- Natalia Luzuriaga
- Isaac Milarsky
- Patrick Sier
- Alberto Colon
- Cassie Rubio
- Yuyang Zhong

