GSOC 2022 Proposal

Cuneiform Digital Library Initiative

Seals Portal

Name	:	Aditi Singh
Email	:	aditiansingh@gmail.com
GitLab	:	aditiansingh
GitHub	:	Aditi-Singh16
Location	:	Mumbai, India
Timezone	:	UTC+5:30

Abstract -

The seals portal aims at displaying all of the resources in the CDLI concerning a seal, aggregated in one place, with important features like viewing image annotations and graphical representation of seals chemistry. The portal lists different CDLI seals as groups i.e. physical, composite, sealings, and all cdli seals for a better user experience. Secondly, it shows a list of best-attested composite seals along with an image and description. Lastly, it presents a table containing seals and impressed tablets by period, here the users can view seals grouped by period (name and duration). The seals' single view will display the seal image, a graphical way to present seals' chemical information; this graph can be viewed in a bar chart, pie chart, line chart and can be converted to tabular format. The seals' single view includes accordions to display physical information, inscription, identifiers, provenience, Composites, Inscriptions, and Chemical information. It also displays Image annotations, which will help users to understand and know intricate details about the seal. This project will also have a standalone converter tool for converting annotations from COCO format to w3c.json format. There will be a bulk update interface to upload annotations to the database the user can upload the annotations in w3c.json or COCO format.

Why Seals Portal?

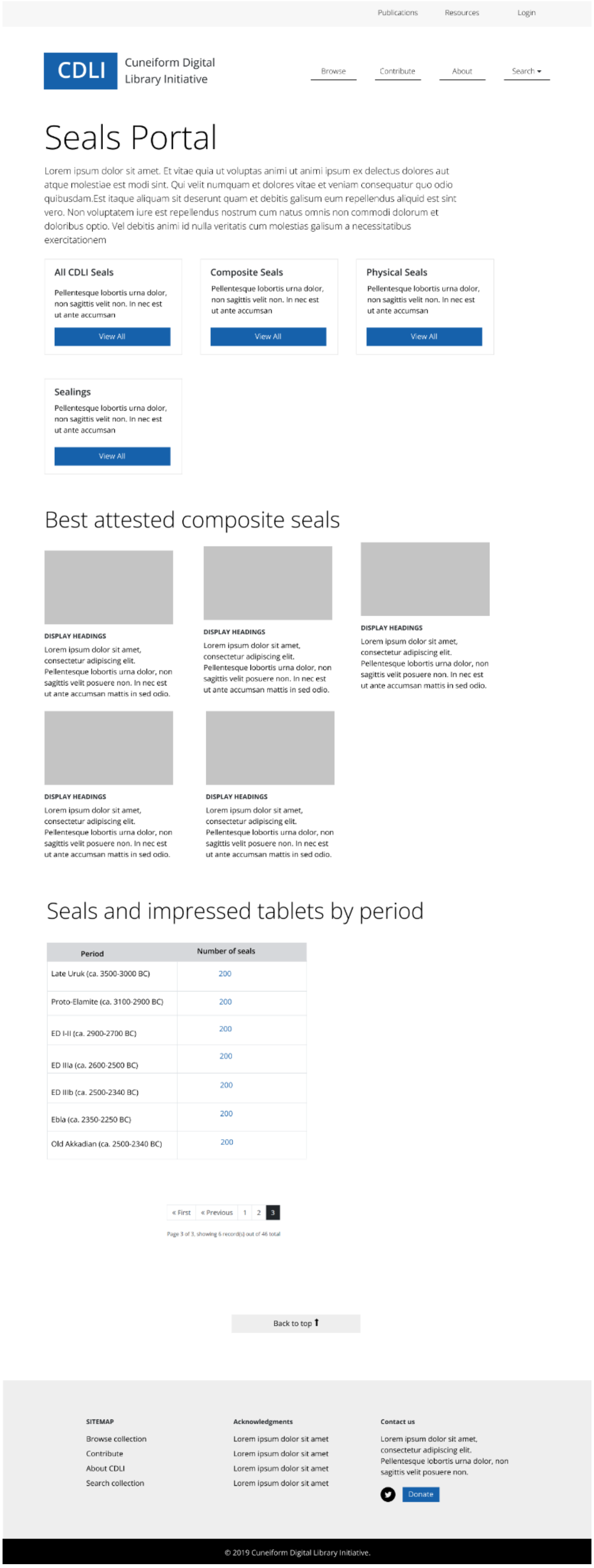
The Seals portal will be helpful for students or professionals learning, teaching, and researching ancient Mesopotamian seals.
This portal will contain all information about seals gathered in one place while there is no such portal existing yet. The seal single view will present a graphical representation of chemical elements of the seal vs their amount detected and a tabular representation of seal chemistry along with seal image annotation. These visualizations would give a better understanding of the chemical elements of the seal. The converter tool will be helpful as professionals can annotate an image using any tool of their choice and export it in COCO format and those annotations can be easily uploaded and viewed on the CDLI framework once converted to w3c.json format.

Proposed Deliverables -

1. The Seals Portal

The seals portal is the main page or index to access information concerning seals in the CDLI collection (link to Figma design)

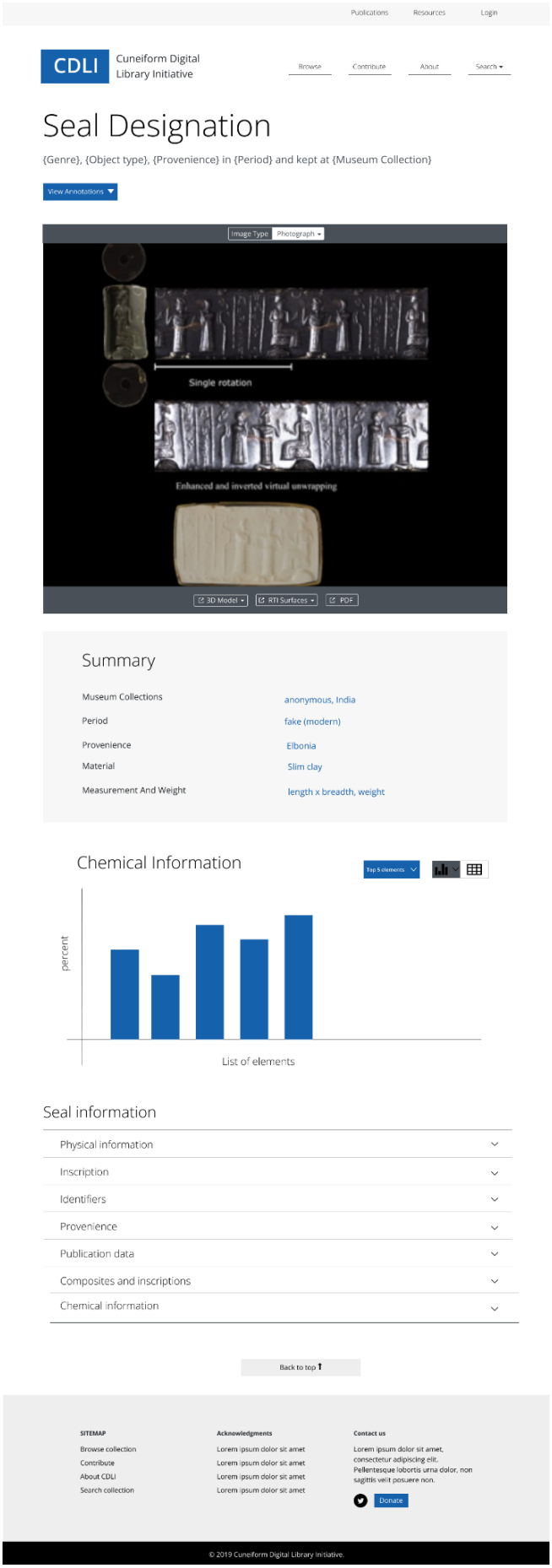
- Displaying and grouping seal categories:
  - Physical Seals
  - Composite Seals
  - Sealings
  - All CDLI seals
- Displaying Best Attested Composite seals with image and description.
- Tabular representation of seals, categorized by period.


Publications
Resources
Login
CDLI
Cuneiform Digital Library Initiative
Browse
Contribute
About
Search
Seals Portal
Lorem ipsum dolor sit amet. Et vitae quia ut voluptas animi ut animi ipsum ex delectus dolores aut atque molestiae est modi sint. Qui velit numquam et dolores vitae et veniam consequatur quo odio quibusdam.Est itaque aliquam sit deserunt quam et debitis galisum eum repellendus aliquid est sint vero. Non voluptatem iure est repellendus nostrum cum natus omnis non commodi dolorum et doloribus optio. Vel debitis animi id nulla veritatis cum molestias galisum a necessitatibus exercitationem
All CDLI Seals
Pellentesque lobortis urna dolor, non sagittis velit non. In nec est ut ante accumsan
View All
Composite Seals
Pellentesque lobortis urna dolor, non sagittis velit non. In nec est ut ante accumsan
View All
Physical Seals
Pellentesque lobortis urna dolor, non sagittis velit non. In nec est ut ante accumsan
View All
Sealings
Pellentesque lobortis urna dolor, non sagittis velit non. In nec est ut ante accumsan
View All
Best attested composite seals
DISPLAY HEADINGS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non. In nec est ut ante accumsan mattis in sed odio.
DISPLAY HEADINGS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non. In nec est ut ante accumsan mattis in sed odio.
DISPLAY HEADINGS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non. In nec est ut ante accumsan mattis in sed odio.
DISPLAY HEADINGS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non. In nec est ut ante accumsan mattis in sed odio.
DISPLAY HEADINGS
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non. In nec est ut ante accumsan mattis in sed odio.
Seals and impressed tablets by period
Period
Number of seals
Late Uruk (ca. 3500-3000 BC)
200
Proto-Elamite (ca. 3100-2900 BC)
200
ED I-II (ca. 2900-2700 BC)
200
ED IIIa (ca. 2600-2500 BC)
200
ED IIIb (ca. 2500-2340 BC)
200
Ebla (ca. 2350-2250 BC)
200
Old Akkadian (ca. 2500-2340 BC)
200
« First
« Previous
1
2
3
Page 3 of 3, showing 6 record(s) out of 46 total
Back to top
SITEMAP
Browse collection
Contribute
About CDLI
Search collection
Acknowledgments
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Contact us
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non.
Donate
© 2019 Cuneiform Digital Library Initiative.

2. The single seal's view

The single seal's view is the template page that will display all relevant information concerning a single seal (physical or composite). It is based on the single artifact view, but I have adapted it to fit the requirements which are specific to seals.
Implementation of a JS library for displaying image annotations. (Link to Figma design)

- Implementation of a PHP library for graphical representation of chemical information of seals.
    - A feature to view only top n chemical elements (based on amounts detected)
    - A switch to convert the graph into a table view.

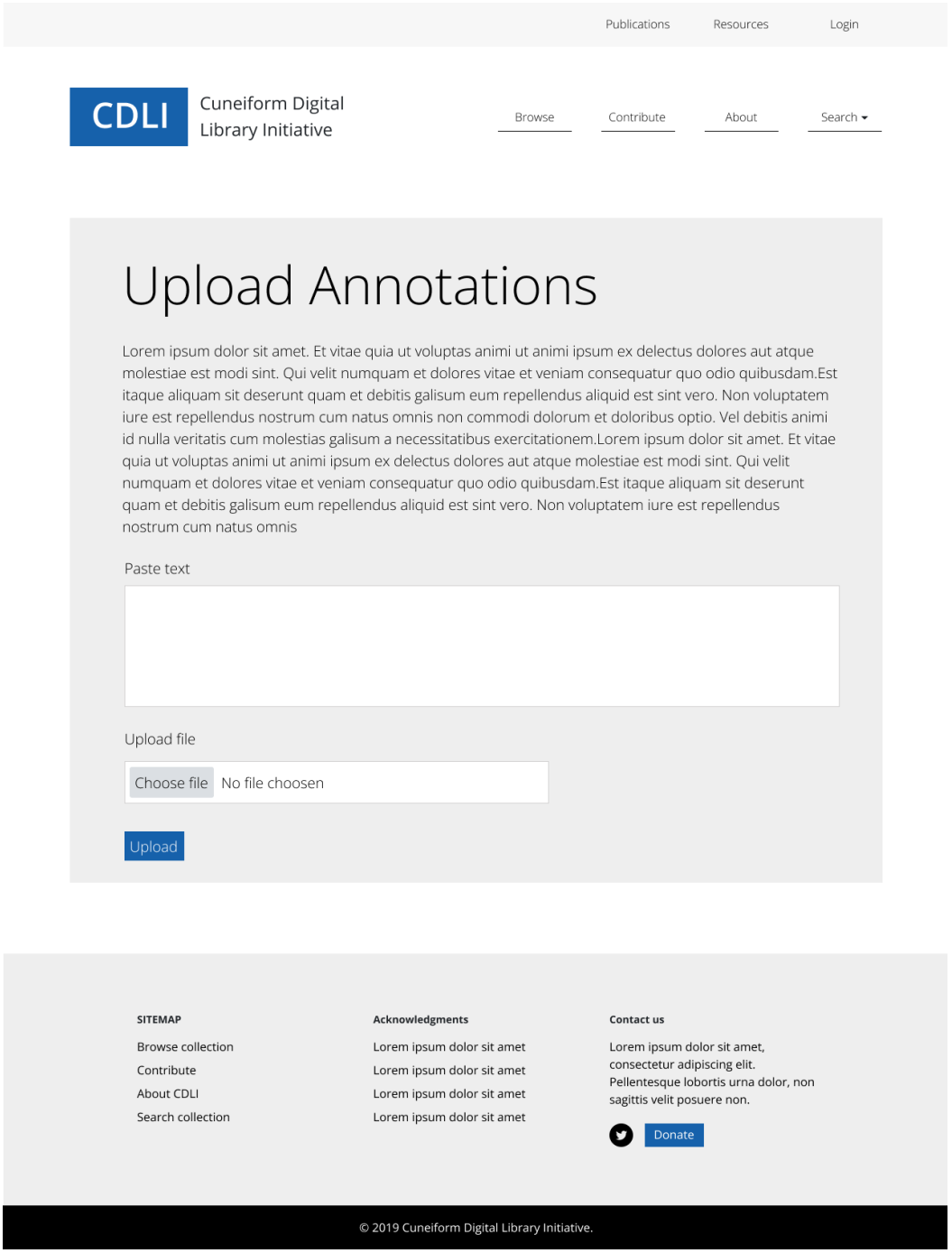

Publications
Resources
Login
CDLI
Cuneiform Digital Library Initiative
Browse
Contribute
About
Search
Seal Designation
{Genre}, {Object type}, {Provenience} in {Period} and kept at {Museum Collection}
View Annotations
Image Type
Photograph
Single rotation
Enhanced and inverted virtual unwrapping
3D Model
RTI Surfaces
PDF
Summary
Museum Collections
anonymous, India
Period
fake (modern)
Provenience
Elbonia
Material
Slim clay
Measurement And Weight
length x breadth, weight
Chemical Information
Top 5 elements
percent
List of elements
Seal information
Physical information
Inscription
Identifiers
Provenience
Publication data
Composites and inscriptions
Chemical information
Back to top
SITEMAP
Browse collection
Contribute
About CDLI
Search collection
Acknowledgments
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Contact us
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque lobortis urna dolor, non sagittis velit posuere non.
Donate
© 2019 Cuneiform Digital Library Initiative.

3. Converting COCO annotations to W3C annotations format -

   There are different manual image annotation tools, for example, VIA which exports annotations in COCO format and we will be using annotorious to represent image annotations to users, thus we need a converter from coco annotations to w3c.json format.

4. Bulk update of Image Annotations Table

   An Interface to upload annotations, convert it to w3.json format if it's in COCO format, and add the annotation to the image_annotations table.

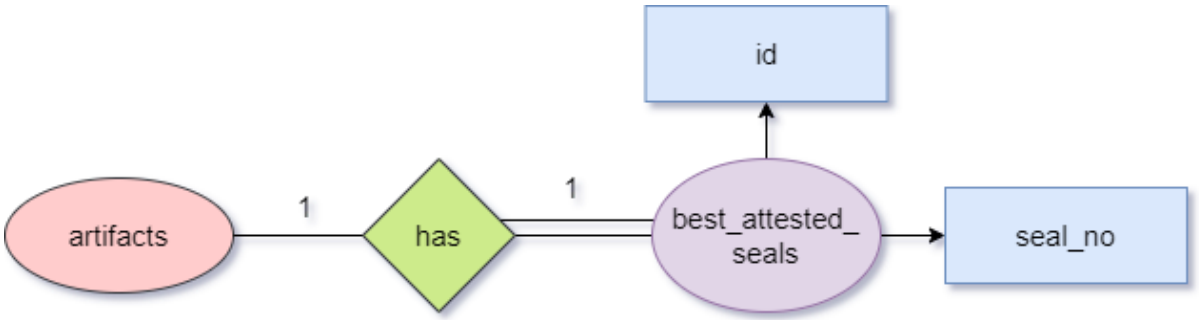


Milestone Details-

1. The seals portal

Displaying Best Attested Composite seals with images and descriptions.

Database table -
BestAttestedSeals (id, seal_no, artifact_id)



Adding the relation to model files

```
$this->hasOne('BestAttestedSeals', [
    'foreignKey' => 'artifact_id'
]);
```

Using it in the controller.

```
//for the best attested seal cards
$bestseals = $this->Artifacts->find('all',['contains'=>'BestAttestedSeal']);
```

Tabular representation of seals, categorized by period.

Seals and impressed tablets by period

Period	Number of seals
Late Uruk (ca. 3500-3000 BC)	200
Proto-Elamite (ca. 3100-2900 BC)	200
ED I-II (ca. 2900-2700 BC)	200
ED IIIa (ca. 2600-2500 BC)	200
ED IIIb (ca. 2500-2340 BC)	200
Ebla (ca. 2350-2250 BC)	200
Old Akkadian (ca. 2500-2340 BC)	200

Here, the number of seals is the number of seals in that particular period and provenience.

The number of seals will link to a page showing seals and impressed tablets with filters.
for example, the search link for a seal with period = Late Uruk(ca. 3500-3000) and provenience = Uruk (mod. Warka) would look like this

- http://127.0.0.1:2354/search/1648315949?keyword=india&addedFilters=Late%20Uruk(ca.%203500-3000),Uruk%20(mod.%20Warka)

I will be using the existing search page display to display this.

For this, I am working on an issue for creating a share search link with filters that will show the exact page to the user with all the filters applied.
Issue link - Sharing search page link with filter
PR link - #584

2. The Single Seals View.

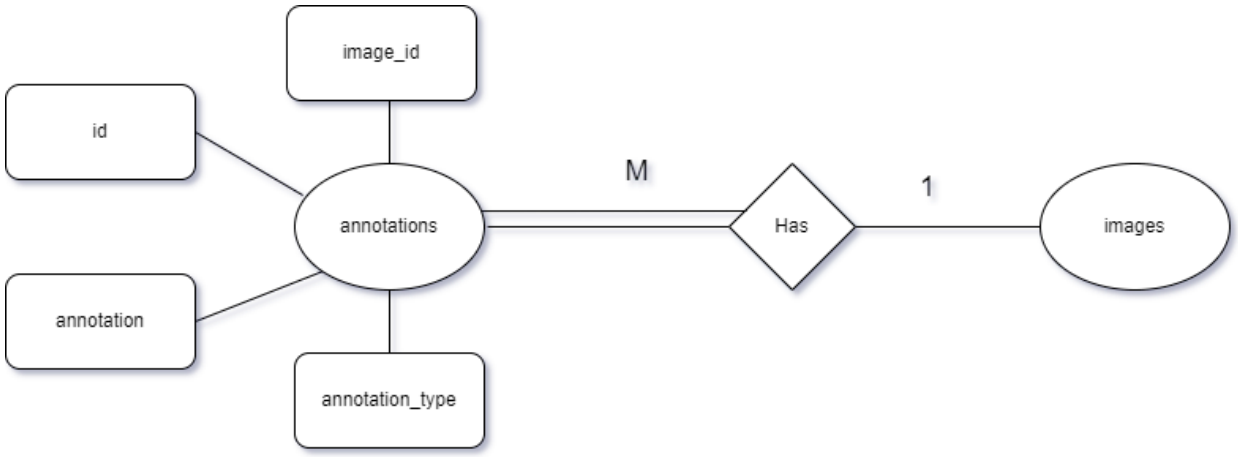
Implementation of a JS library for displaying annotations.

- For annotations, I will be using the Annotorious OpenSeadragon library.
- Setup - I will be adding the script tag in the required ctp file.

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/@recogito/annotorious-openseadragon@latest/dist/annotorious.min.css">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/openseadragon/2.4.2/openseadragon.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@recogito/annotorious-openseadragon@latest/dist/openseadragon-annotorious.min.js"></script>
```

To load annotations from DB:

The annotations will be of the format w3.json
Annotations table:-



When the user clicks on the view annotations button this function will be called

1. First, we will create the required model files for the annotations table.
2. Add annotations table as an association in the ImagesTable.

```
$this->belongsTo('Annotations', [
            'foreignKey' => 'annotation_id',
        ]);
```

```
$annotations = TableRegistry::get('Images')->find('all')
                ->contain(['Annotations', 'Artifacts'])
                ->where([
                        'Annotations.image_id'=>$imageId,
                        'Annotations.annotation_type'=>$type,
                        'Artifacts.id'=>$id
                ]);
```

Viewing annotation

Html:

```
<div id="imageId" style="width: 800px; height: 600px;"></div>
```

JS:

```
var viewer = OpenSeadragon({
                id: "sealImg",
                navigatorSizeRatio: 0.25,
                showNavigator: true,
                prefixUrl: "images",
                tileSources: "path to dzi file"
        })
 var anno = OpenSeadragon.Annotorious(viewer);
 anno.setAnnotations(/* Annotation Array */)
```

Creating tiles using deepzoom:

When user clicks on View annotations, we will be creating image tiles first using deepzoom and gd package.

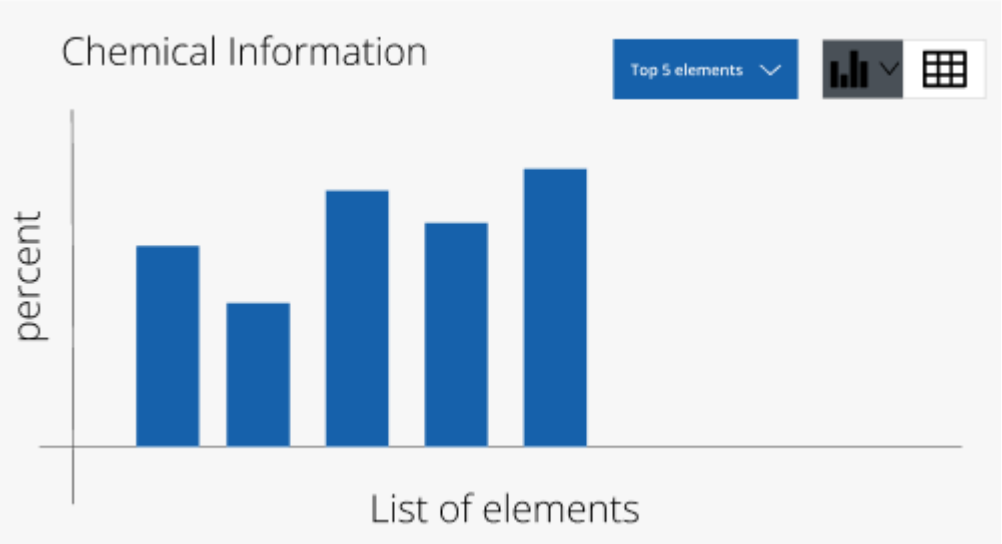
This will generate a .dzi file which will be the tile source

I have been facing difficulties with installing gd extension to the php.ini files in the cdli_dev docker container, but I will make sure to find a solution to this.

```
$deepzoom = DeepzoomFactory::create([
                'path' => 'images', // Export path for tiles
                'driver' => 'gd',
                'format' => 'jpg',
        ]);
 $response = $deepzoom->makeTiles(/*path of image*/);
```

Graphical representation of chemical information of seals.

- The user has three options
  - To view top 5,10 and All elements
  - View the chemical information in tabular format
  - View chemical information in graphical format
    1. Bar graph
    2. Line graph
    3. Pie chart



1. First, we will get the JSON data from DB and then loop through it to store the element name in the Y-axis array and the value in the X-axis array.
2. I will be using HighCharts Library which is free and open source.

```
$dataXAxis = array();
$dataYAxis = array();
$seals = $this->Seal->find()
        ->where([
            'Seal.id = ' => $id
         ]);
foreach($seals as $seal){
     $labelArr =  explode('label":',$seal->jsondata);
     $valueArr =  explode('value":',$seal->jsondata);
     for ($i=1;$i<count($labelArr);$i++) {
        $valueele = explode("},",$valueArr[$i]);
```

```
            $labelele=explode(",",$labelArr[$i]);
        array_push($dataXAxis,str_replace('"','',$labelele[0]));
             array_push($dataYAxis,(int)str_replace("}
        ]",'',$valueele[0]));
            }
        }
        $chart = new Highchart();
        $chart->chart->renderTo = "container";
        $chart->chart->type = "column";
        $chart->title->text = "Seals Chemistry";
        $chart->xAxis->categories = $dataXAxis;
        $chart->yAxis->min = 0;
        $chart->yAxis->title->text = "Amount";
        $chart->tooltip->formatter = new
        HighchartJsExpr("function() {
                    return '' + this.x +': '+ this.y
        +'{unit}';}");
        $chart->plotOptions->column->pointPadding = 0.2;
        $chart->plotOptions->column->borderWidth = 0;
        $chart->credits->enabled = false;
        $chart->series[] = array(
              'name'=> 'Elements',
              'data' => $dataYAxis
        );

$chart->printScripts();
```

A feature to view only top n elements (based on weight)

```
if(isset($n)){
   arsort($dataAll);
   $dataAll = array_splice($dataAll,0,$n);
   $dataXAxis = array_keys($dataAll);
   $dataYAxis = array_values($dataAll);
}
```

Where $dataAll = an object with keys as element name and value as element weight.

We can easily convert it to line chart by changing the type to ‘line’
To convert it to pie chart we can follow the following approach

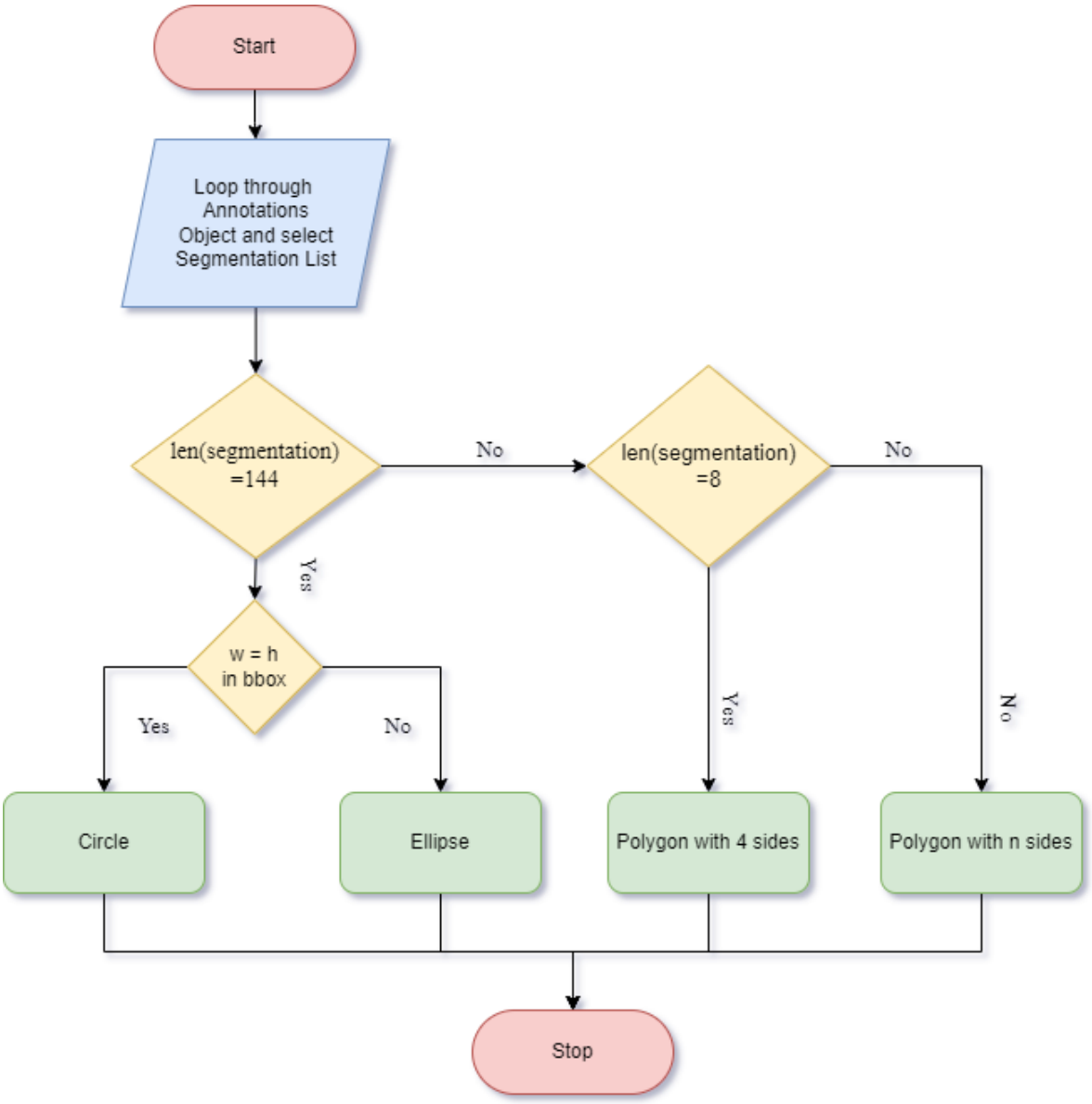
```
$chart->plotOptions->pie->allowPointSelect = 1;
$chart->plotOptions->pie->cursor = "pointer";
$chart->plotOptions->pie->dataLabels->enabled = 1;

$chart->plotOptions->pie->dataLabels->formatter = new
HighchartJsExpr(
                "function() {
                    return '' + this.point.name +': '+ this.y +' gm';
}");
$dataPieChart=array();
for($i=0;$i<count($dataXAxis);$i++){
array_push($dataPieChart,array($dataXAxis[$i],$dataYAxis[$i]));
}

$chart->series[] = array(
     'type' => "pie",
     'name' => 'Elements',
     'data' => $dataPieChart
 );
```

3. Implementing a convertor from coco annotations to w3c.json format

An annotation object in coco annotation has the following parts:

```
{
   "info": {...},
   "licenses": [...],
   "images": [...],
   "annotations": [...],
   "categories": [...]
}
```

Flowchart to detect the shape and its coordinates



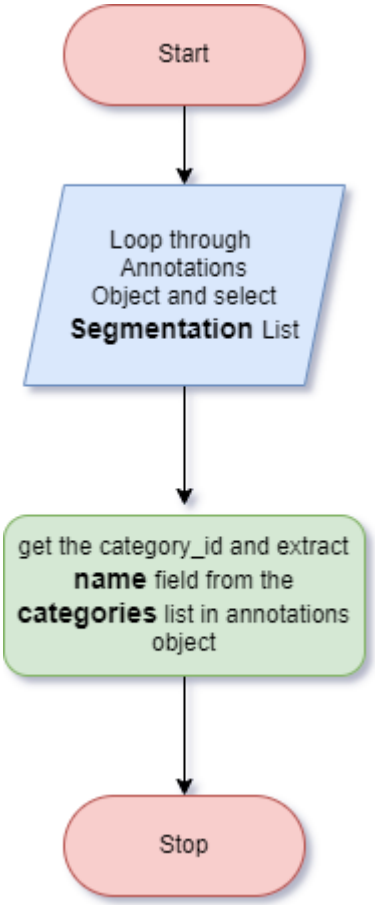
We have the points for SVG plotting, thus, we can fill in the annotations in w3c.json format.

To get the annotation text we can use the following approach -

The categories list in the COCO annotations object contains the following fields

```
{
        "supercategory":"",
        "Id": 1,
        "name":  ""
}
```

Here, the supercategories can be based on the type of annotations



4. Bulk update of image_annotations table

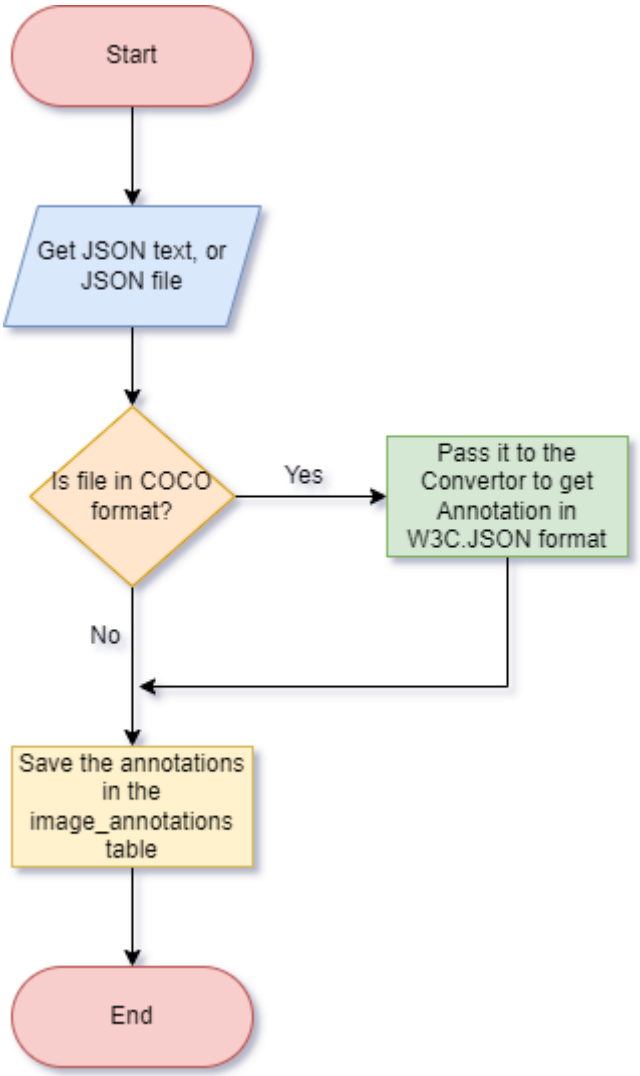
The user will upload the JSON file or paste the JSON in the text box.
The content could be of this format

{
“artifact_id”:” P number of the artifact”,
//Annotation in W3C JSON or COCO format
}

For more than one artifact, we can have an array of JSON.
Example -

[
{“artifact_id”:” P number of the artifact”, Annotation in W3C JSON or COCO format},
{“artifact_id”:” P number of the artifact”, Annotation in W3C JSON or COCO format}
]

Flowchart of add() function in Annotations Controller.



Timeline

I will be denoting 3-4 hrs every day and an additional 2 hrs every week for documentation.

Milestone	Week	Objective(s)	Tasks
Milestone 1	Week 1 13 June - 19 June Week 2 20 June - 27 June	Implement required changes to the database and edit the model files accordingly	- Setting up the image_annotations table - Creating respective model files - Bulk update of image_annotations table
	Week 3 28 June - 4 July	Testing Milestone 1 manually	

Milestone 2	Week 4
	5 July - 11 July
	Week 5
	12 July - 18 July
		The seals portal	- Displaying Best Attested Composite seals with image and description.
			  - Create a table and respective model files for best_attested_seals
			- Tabular representation of seals, categorized by period.
			- Displaying and grouping seals categories:
			  - Physical Seals
			  - Composite Seals
			  - Sealings
			  - All CDLI seals

	Week 6 19 July - 25 July	Testing Milestone 2 manually	
Milestone 3	Week 7 26 July - 1 Aug Week 8 2 Aug - 8 Aug Week 9 9 Aug - 15 Aug	Implementing seals single view	- Implementation of a PHP library for graphical representation of chemical information of seals. - A feature to view only top n elements (based on weight)

			- A switch to convert the graph into a table view. - Graphical visualisation of element weight in that seal in the form of pie chart, line chart, and bar graph
	Week 10 16 Aug - 22 Aug	Testing Milestone 3 manually	
Milestone 4	Week 11 23 Aug - 29 Aug	Adding image annotation viewer and integrating it with annotorious -open seadragon	- Create the viewer page - Generate tiles for deepzoom - Integrate with annotorious-openseadragon
	Week 12 30 Aug - 5 Sept	Testing Milestone 4 manually	
Milestone 5	Week 13 6 Sept - 12 Sept	Implement a convertor from coco format to w3c.json format	- To Study more about w3c.json and COCO image annotations format - Comparing both coco and w3c.json annotation formats to create SVG points layout - Writing a python script that takes in coco_format and gives w3c.json format - Add the convertor to the docker container.
	Week 14 13 Sept - 19 Sept	Testing Milestone 5 manually.	

Contributions to CDLI -

I have been contributing to the CDLI framework since august 2021

1. Composite scores and index - feature
2. Populate parsed fields of inscription table - feature
3. Composite witness section of artifacts single view - feature
4. Highlights and news page - feature
5. Added deceased fields in Author index, add and edit forms - feature
6. Fixed ordering issue in CDLJ - Improvement
7. Fixed redirection to current URL after login - Improvement
8. Collections Index Improvement - Improvement

All the contributions can be viewed here.

Why you?

I have been contributing to CDLI since August 2021, and have solved issues with good first tags on them to issues with the highest priority and advanced tech stack. This has helped me to gain a good understanding of the cdli database, the folder structure, and the CakePHP framework.
I would be a good fit for this project since I have experience working with artifacts, and inscriptions fields, and have done an ample amount of research related to image annotations.
The opportunity to contribute to this project would be very beneficial as it will help me showcase my full-stack development skills and enhance my coding practices.

Past Projects -

I have done several full-stack development projects.
All of my projects can be viewed here - List of Projects
The tech stacks I have worked on are

Front-End	ReactJS, NextJS, HTML, CSS, JavaScript
Back-End	MongoDB, MySQL, PHP, NodeJS, Python
ML	Done project on classification Algorithm.
Hosting Platforms	Heroku and WebHost000

Work Experience

1. Software Developer Intern - Star Union Daichi Life Insurance
    - Contributing to Data Validation across different insurance policies

2. App development Internship – Swechchha
   Swechchha is a project by the Municipal Corporation of Greater Mumbai. As an intern, my task was to develop an application that will help them to -
    - Manage and maintain community toilets
    - Raise complaints about them
    - Create channels for different issues and discuss them
    - Direct chats among users

3. AWS intern at CAPGEMINI
    - I was trained in AWS concepts, during the starting weeks of my internship
    - I developed a Reusable utility project using AWS pinpoint service and Flutter
    - With this, we can send SMS to any mobile number with a particular message template.

4. Full-stack Development internship at Swagit
   Google play store link - SWAGIT
    - UI and backend for setting user preferences.
    - Created the backend and UI for the Diwali contest.

- Adding google analytics, in-app messaging, and dynamic links to the app and website.
- Created celebrity style decode page feature.
- Worked on creating the IOS build of the app and UI enhancement Worked on the stylist interface

Post-GSOC Work -

I will be working on making the annotations searchable by extending the search query. I will continue contributing to the CDLI framework and work on adding features that will make the seals portal better.