Google Summer of Code

Journals Management & Display

Google Summer Of Code 2020

Nisheal A John (@Nishealj)

Karunya University

nisheal.work@gmail.com

Hi, I'm **Nisheal John,** a final year student at Karunya University Coimbatore pursuing my bachelor's degree in Computer Science and Engineering.

I have always manifested a strong academic record and nurtured with zeal for learning, I've never failed to miss the opportunity to learn more.

Thank you for giving me an opportunity to contribute to your organization through Google Summer of Code.

Prior Experience :

I've developed various projects in MERN Stack and php frameworks (cakePHP, Laravel), **SAFEROADSPROJECT** is one of the notable projects among those :

- **SAFEROADS PROJECT** is a software product built in Laravel to contribute towards road safety, it provides a dedicated environment for management of vehicle and tracks driving reports .

I've also worked with :

- **4S Information Technology ,** in Ankara Turkey where I worked in deploying critical software/network security systems.
- **MERN STACK COMMENT BOILERPLATE,** to provide a plugable comments interface for mern-stack applications.

Contributions to cdli :

I've contribution in the cdli in the areas of bugfix and improving dev-workflow,

- Improvements in developer workflow, added lint checks for SCSS, CSS & PHP #76, #74
- Added GitLab pipeline for CI/CD #45, #51
- General improvements & bugfix #36, #55, #72

Here is a complete list of contributions made by me

1. Project Overview :

Journals Management & Display

This project focuses on the publication of various types of journals which CDLI hosts, currently CDLI hosts four journals, two peer-reviewed journals (CDL Journal and CDL Bulletin), the CDL Pre-prints repository and the CDL Notes. For the upload of journals, appropriate designs and workflow need to be developed which includes the development of required routes, controllers & views. Adding to the above project necessary end-to-end & unit testing phases should be established to improve the development workflow and avoid build issues.

2. Timeline Overview :

- **Mar 16 to 17 May '20 (9 Weeks)**

  This duration includes the pre-GSoC phase and the community bonding period, which will be utilized to improve project timelines, building initial phases and Learning more about various other projects & teams of CDLI.

- **May 18 to August '17 (13 Weeks)**

  1. After the **Milestone 1** : All the necessary **routes**, **controllers, models** & **views** will be developed, this includes the **CRUD** operations in the admin dashboard for articles**.**

  2. After the **Milestone 2** : All the articles will be available for users (**Display Of Journals**) and necessary authentication will be added

  3. After the **Milestone 3 :** Addition of **unit testing** for the necessary **CURD** and **display** operations of the articles.

3. Milestone Details:

3.1 About Milestone 1 :

CRUD operations for articles in Admin Dashboard.

This milestone includes the development of a articles upload dashboard where the editor can upload an article (CDLJ, CDLN, CDLP & CDLB see **appendix-1** to know more on types of articles)

Abbr	Name	Link
CDLB	Cuneiform Digital Library Bulletin	View here
CDLJ	Cuneiform Digital Library Journals	View here
CDLN	Cuneiform Digital Library Notes	View here
CDLP	Cuneiform Digital Library Preprints	View here

(Appendix 1 : Types of articles*)*

The 4 articles of CDLI will have different upload & edit pages to avoid confusion and provide easy routing, example

```
/admin/journals/add/cdlj -> Add new CDLJ Journal
/admin/journals/edit/cdlj/{id} -> Edit {id} CDLJ

/admin/journals/add/cdlp -> Add new CDLP Journal
/admin/journals/edit/cdlp/{id} -> Edit {id} CDLP
```

While adding an article, the necessary fields are article **title**, **authors** and **content** where article title & author are not complex, it will be a general textbox to enter title and authors name.

For the CDLJ & CDLB, editor will be given a textarea input for the article content and a file upload option to upload the appropriate pdf or latex file of the article, which they created at the overleaf or similar website and the article dashboard will extract the content from the pdf(any type of pdf including latex) or the latex file then convert it into html and save into the database as the article content.

For CDLP, editors can upload the pdf of the publication and it will be available for the users to download on the CDLP page. Conversion to HTML is not needed.

For CDLN, editors will upload the content using the CKEditor which will be stored in the database and rendered for the users where they can download it as a pdf.

Name	Type	Structure
CDLB	Bulletin	Pdf uploaded by the editor and converted by the dashboard
CDLJ	Journal	Pdf uploaded by the editor and converted by the dashboard
CDLN	Notes	Pdf generated from the html content input from the editor
CDLP	Preprints	Pdf uploaded by the editor

(Appendix 2 : Journals structure & upload description)

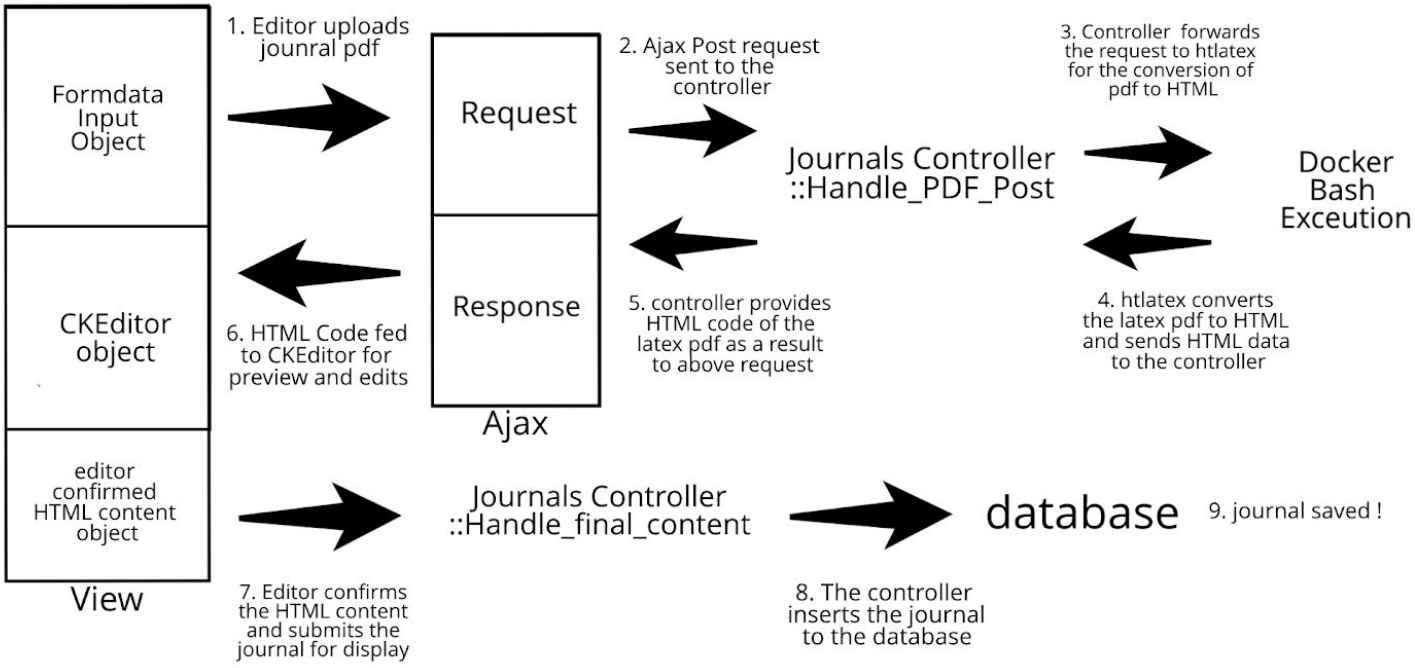
To convert the latex template into html we will be using `htlatex/pdftohtml` which belongs to the family to `tex4ht` and `mactex` , the overall process flow for journal articles upload is given below.

At frontend to editor,

Fig 7a *Fig 7b* *Fig 7c* *Fig 7d*

The possible CRUD operations here are the addition of journal articles to the database, edit of journal articles i.e getting data from database and giving it to CKEditor for editors to edit and then the deletion of the journal articles.

At backend, conversion of latex template to html content.



The above process flow is how the journal articles latex pdf or latex file will be converted into the html content and saved to the database, further adding any more additional details of the journal articles can be easily added using the the controllers and I don't find it a challenge, also the temporary design implementation of the same can be found at the bottom section-7 (source and figures),

In general, the process flow of adding the journal articles will be very easy and user friendly for the editors. Editor has to input a title, upload the files then verify the conversion (HTML content) and then save it.

Database Migrations (or) Schema Updates,

The database schema for articles table has a few changes and author attributes has many-to-many tendencies hence, appropriate relations should be defined.

Appendices 3 articles *database schema*

articles- id	title	author	content	file	type	status	created_at	updated_at
int(PM)	Lontext	Lontext	Longtext	Longtext	enum(CDLJ,CDLN,CDLP & CDLB)	enum(0,1)	Timestamp	Timestamp

Appendices 4 Authors database schema

author-id	author_firstName	author_lastName	author_details	created_at	updated_at
int(PM)	Lontext	Longtext	Longtext	Timestamp	Timestamp

*Appendices 5 Author-to-*articles *database schema*

id	author_id	journal_id	created_at	updated_at
int(PM)	int(FK)	int(FK)	Timestamp	Timestamp

The above are the three tables for the journals management where the author-to-articles is the table providing m-to-m relations for the authors to articles, author-to-articles has foregin key articles-i'd and author-id hence for each article having n authors, n entries will be mapped at this table.

3.2 Challenges in milestone 1

- **Formatting of journal articles:**

  We should take care of the journal articles formatting. we have to create the best expected html content of the latex pdf or latex, if any custom header is applied which is similar to all the journal articles it can be initialized as constant i.e if we can follow only one template for all the journal articles we will make it a constant(hardcoded) footer/header, so the editors only have to enter the latex body.

- **Handling of images and equation inside a journal articles:**

  We have already created a latex pdf to html converter, latex to html converter and found that both the images and equations are handled as expected, further incase of any issues the images can be linked with an a tag (Relative link) and replace in the pdf and equation can be built in CKEditor with MathJax.

3.3 About Milestone 2 :

Display of articles.

At the end of the first milestone we will have a full fledged dashboard which will be capable of uploading the articles in the desired format to the database with all CRUD operations, in the milestone 2 we will look forward to displaying these articles in the respective pages in the user end.

As we currently have the 4 menus, `{'Publications','Resources','Login','Register'}`

We will add the articles to the publication tab as displayed in the image

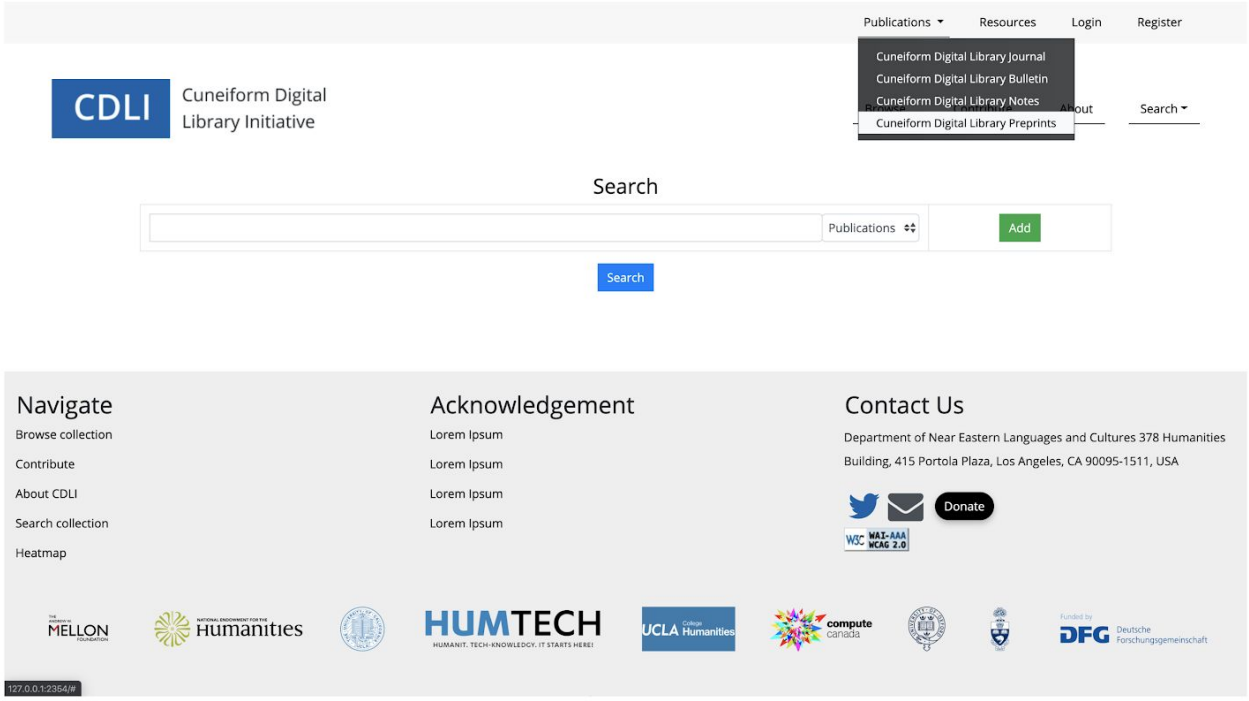


Fig 7f : Menu tab for the display of all types of journals

```
{'Publications' => {  'Cuneiform Digital Library Journals',
                      'Cuneiform Digital Library Bulletin',
                      'Cuneiform Digital Library Notes',
                      'Cuneiform Digital Library Preprints' },
 'Resources','Login','Register'}
```

The structure must be clear with the above json representation, to display the journals will independently route, display the journals in the page foreach journal types

`/journals/cdlj -> view all CDLJ Journal`

`/journals/cdlp -> view all CDLP Journal`

In the above routes, journal will be displayed of each types and will be attached to the respective controllers

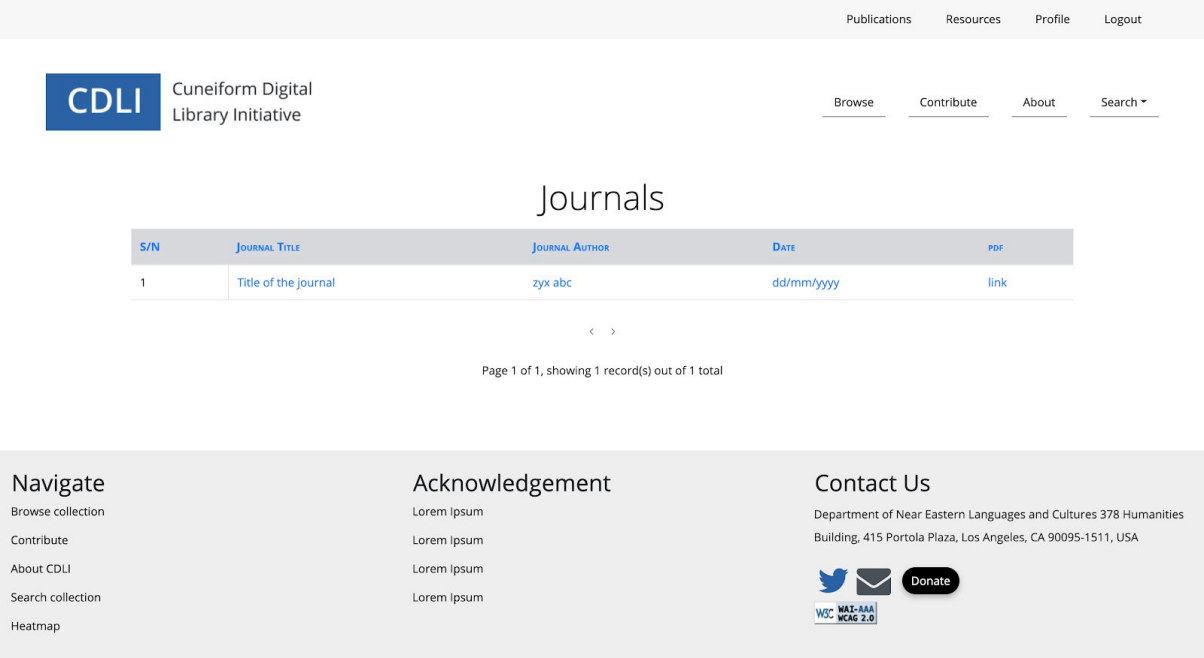
In the display page all the journals details will be listed in the format for the general users

S/N	Journal Title (Linked to html page)	Journal Author(s)	Journal Publication Date	pdf

And in the below format for Admins in the route `/admin/journals`

S/N	Journal Title	Journal Author	Journal Upload Date	Action View/Edit//Delete

See the sample design view below



(*Fig 7e : view of CDLJ journal list for users)*

3.4 Challenges in milestone 2

- **Display of journal html pages :**

  The html content of the journal will be in the database which will be directly displayed to the necessary page route, incase of any common/custom header it will be taken care of with if blocks. You can view the sample journal HTML content at ***Fig 7g : sample view of CDLJ journal HTML***

- **Element preview sanitization to prevent cross-site scripting :**

  Necessary HTML encoding and sanitization needs to be done, to prevent the cross site scripting we will use CakePHP html sanitizers, cake php has the shortcut method of htmlspecialchars as h() functions where the html inputs can be sanitized docs at https://api.cakephp.org/3.4/function-h.html

3.5 About Milestone 3 :

Setting up testing pipelines and enhancements

At the end of the first two milestones we will have a completely functional journal dashboard, we will manually test the entire journal upload/display functionality in this milestone and write the possible unit testing further we will deploy the same to the pipeline.

The needed initial phase-1 for setting up the pipeline is completed at https://gitlab.com/cdli/framework/-/merge_requests/45 in the phase 2, we will make a truncated copy of the current database and make a test-database configuration for the pipeline where we can start the php unit testing, this includes tests for the upload and display controller

We will also try to implement the lint tests when the desired project requirements are complete, we'll improve all the code styles to the latest coding trends and add the lint to the pipeline.

3.6 Challenges in Milestone 3

- **Setting up the test database :**

  The current CDLI database is heavy and has important details, hence we cannot use the same in the pipelines hence, the current database needs to completely truncated and a testing database needs to created

- **Add of testing docs:**

  When the necessary tests are created it is important to make a testing document so that other users can try running the code locally and make changes to the pipelines with confidence.

4. Timeline Brief :

* PR dates are tentative, ** 2 Days buffer after every milestone.				
Milestone	**Week**		**PR Raised***	**PR Merged***
Milestone 1 May 18 - 14 June (4 Weeks)	Week 1 18 May - 24 May Week 2 25 May - 31 May	Implementation of CKEditor and Complete the latex pdf to HTML conversion	28 May	31 May
	Week 3 1 June - 6 June Week 4 7 June - 14 June	Complete insertion of Journal Title. Author and content/HTML to the database	12 June	14 June
Evaluation 1 15 June - 20 June (1 Week)	Week 5 15 June - 20 June	Manually test milestone 1 and finish details for evaluation		
Milestone 2 21 June - 12 July (3 Weeks)	Week 6 21 June - 28 June week 7 29 June - 5 July	Develop routes and controller of listing the types of journals	2 July	4 July
	week 8 6 July - 12 July	Display of HTML Content of the journal (Individually)	10 July	12 July

Evaluation 2 13 July - 18 July (1 Week)	Week 9 13 July - 18 July	Manually test milestone 2 and finish details for evaluation		
Milestone 3 19 July - 9 Aug	week 10 20 July - 26 July week 11 27 July - 2 Aug	Setting up the CI/CD with lint, testing database and a few unit testing for journal dashboard	30 July	2 Aug
	week 12 2 Aug - 9 Aug	Manually test the entire journal publication proposal, build few left out things and ready the project submission	7 Aug	9 Aug
Evaluation 3 10 Aug - 17 Aug	Week 13 10 Aug - 17 Aug	Manually test the project and finish details for final submission		

5. Dev Details :

Devlogs :

I've planned to maintain a google spreadsheet for weekly activity and the work progress where the organization can track my progress and project status.

Timezone :

I'll be in Indian Standard Time (IST) (UTC + 05:30) throughout the project timeline.

Communication :

- ### Contact Information :

Nisheal A John

Email : nisheal.work@gmail.com

Phone : (+91) 8072137030

6. Appendices :

- ***Appendices 1: Journals database schema***

Abbr	Name	Link
CDLB	Cuneiform Digital Library Bulletin	View here
CDLJ	Cuneiform Digital Library Journals	View here
CDLN	Cuneiform Digital Library Notes	View here
CDLP	Cuneiform Digital Library Preprints	View here

- ***Appendices 2: Journals database schema***

Name	Type	Structure
CDLB	Bulletin	Pdf uploaded by the editor and converted by the dashboard
CDLJ	Journal	Pdf uploaded by the editor and converted by the dashboard
CDLN	Notes	Pdf generated from the html content input from the editor
CDLP	Preprints	Pdf uploaded by the editor

- ***Appendices 3 Journals database schema***

article-id	title	author	content	file	type	status	created_at	updated_at
int(PM)	Lontext	Lontext	Longtext	Longtext	enum(CDLJ,CDLN,CDLP & CDLB)	enum(0,1)	Timestamp	Timestamp

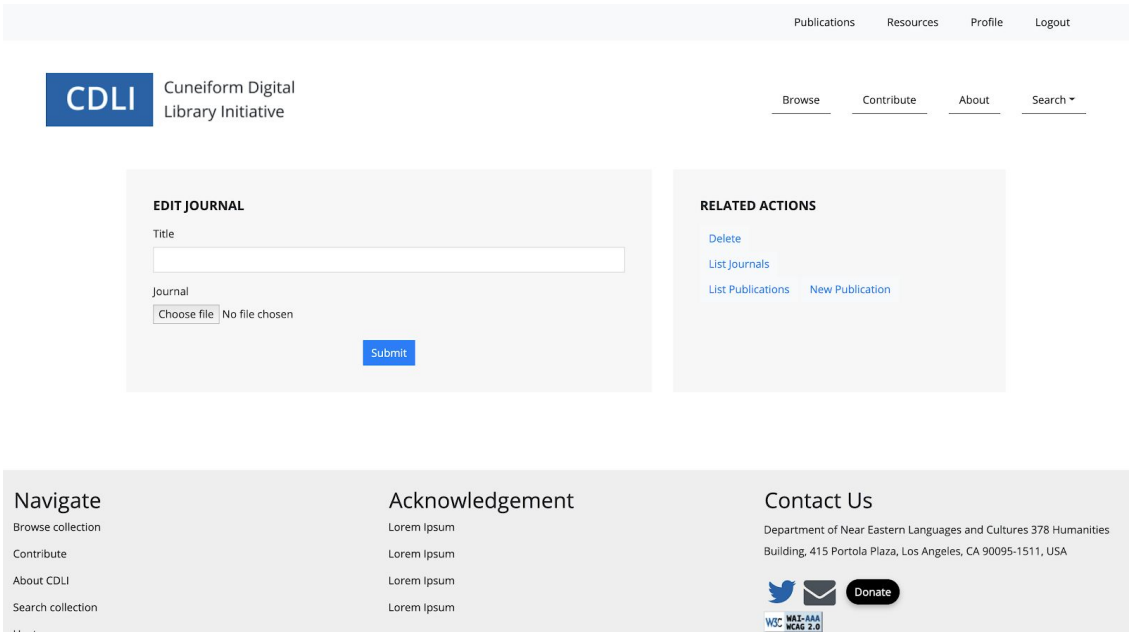
- ***Appendices 4 Authors database schema***

author-id	author_firstName	author_lastName	author_details	created_at	updated_at
int(PM)	Lontext	Longtext	Longtext	Timestamp	Timestamp

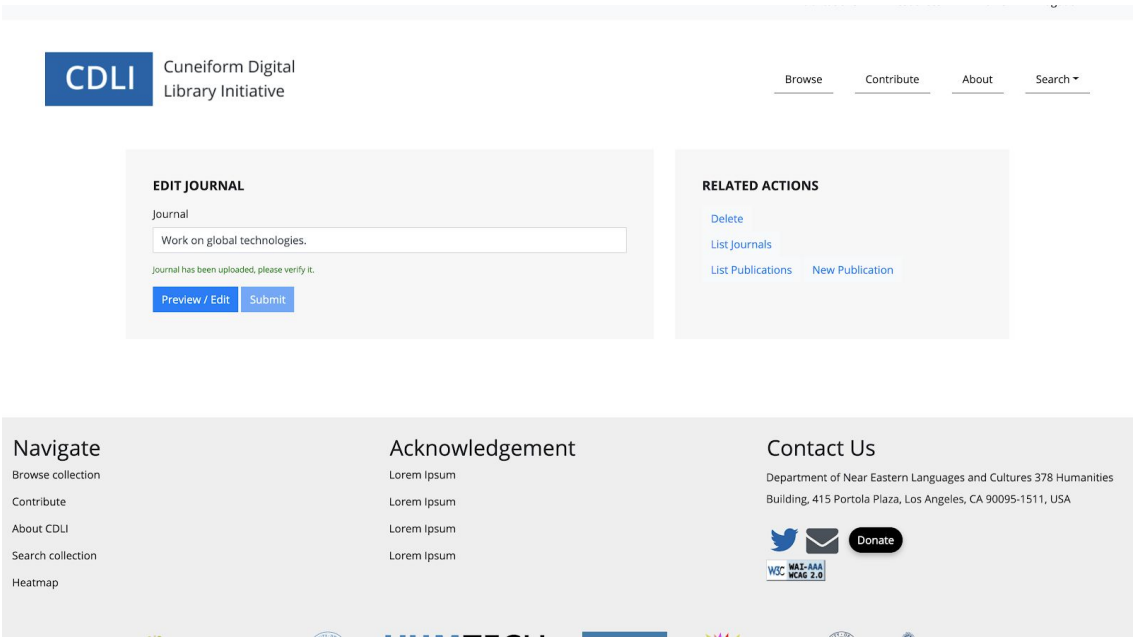
- ***Appendices 5 Author-to-journals database schema***

id	author_id	journal_id	author_order	created_at	updated_at
int(PM)	int(FK)	int(FK)	int	Timestamp	Timestamp

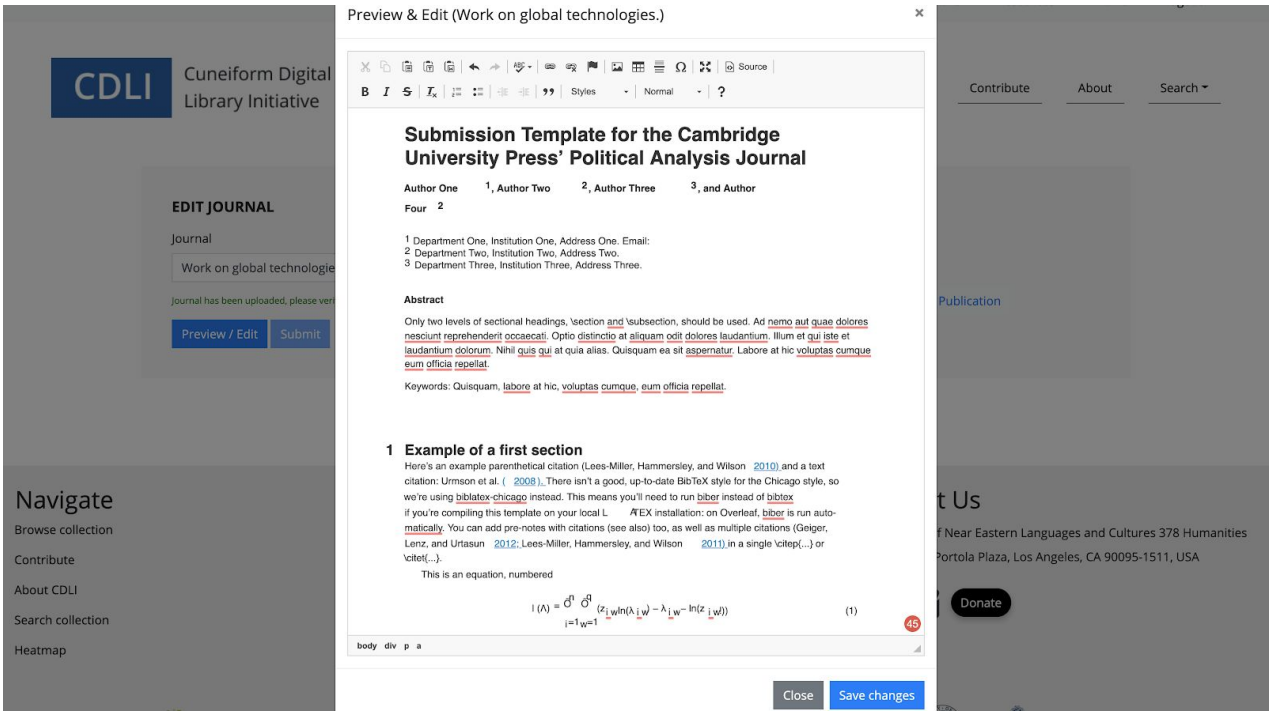
7. Figures & Source :



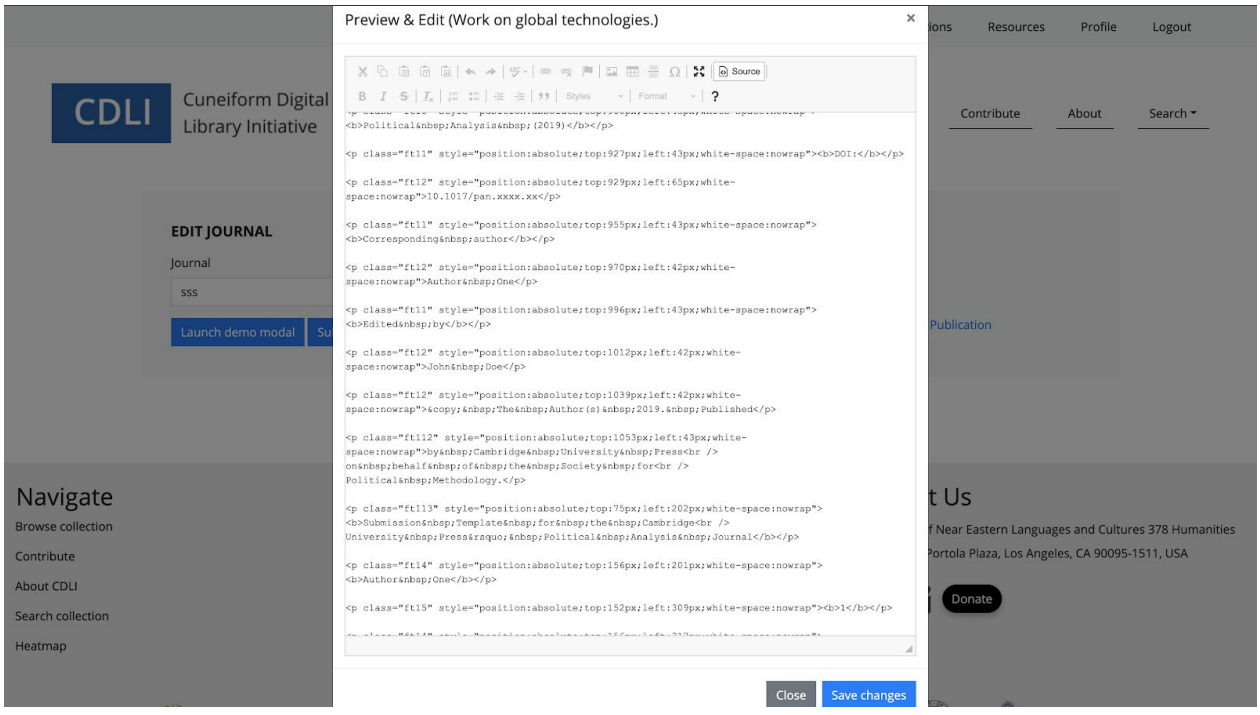
*(**Fig 7a :** Journals upload page t design, **to-be-added:** author textbox & content textarea)*



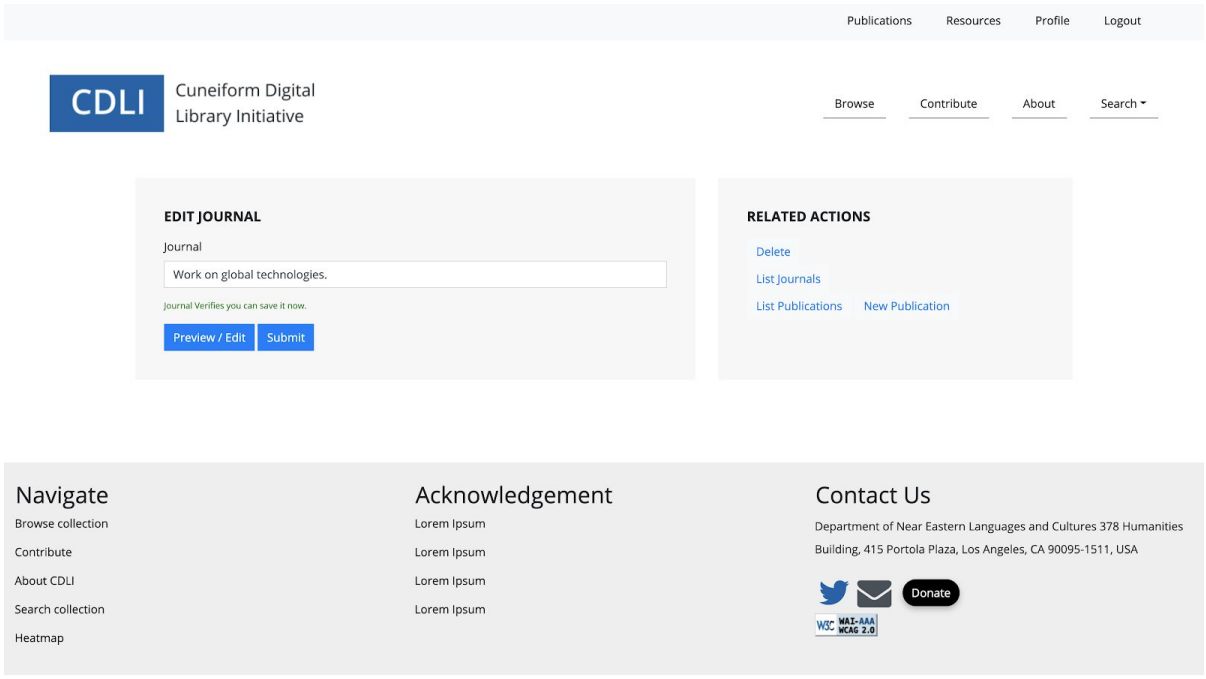
*(**Fig 7b :** After journal's pdf is uploaded, appearance of preview/edit button and disable submit button)*



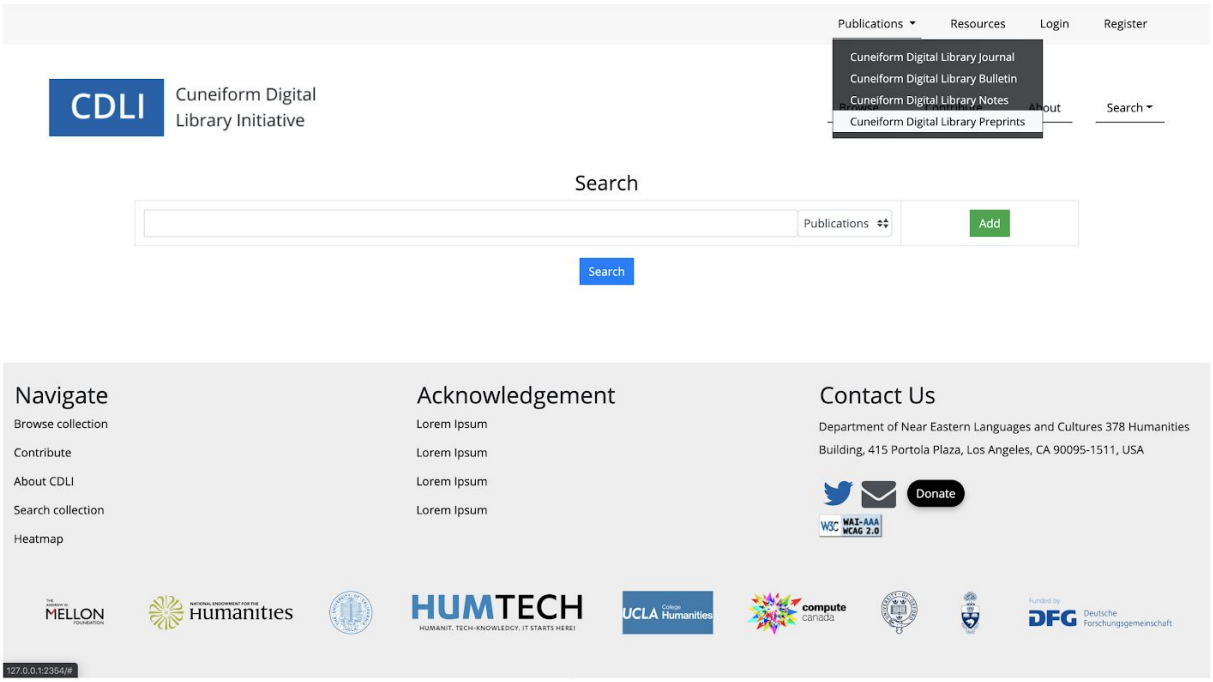
*(**Fig 7c :** when preview/edit journal button is clicked, modal with CKEditor to edit journal)*



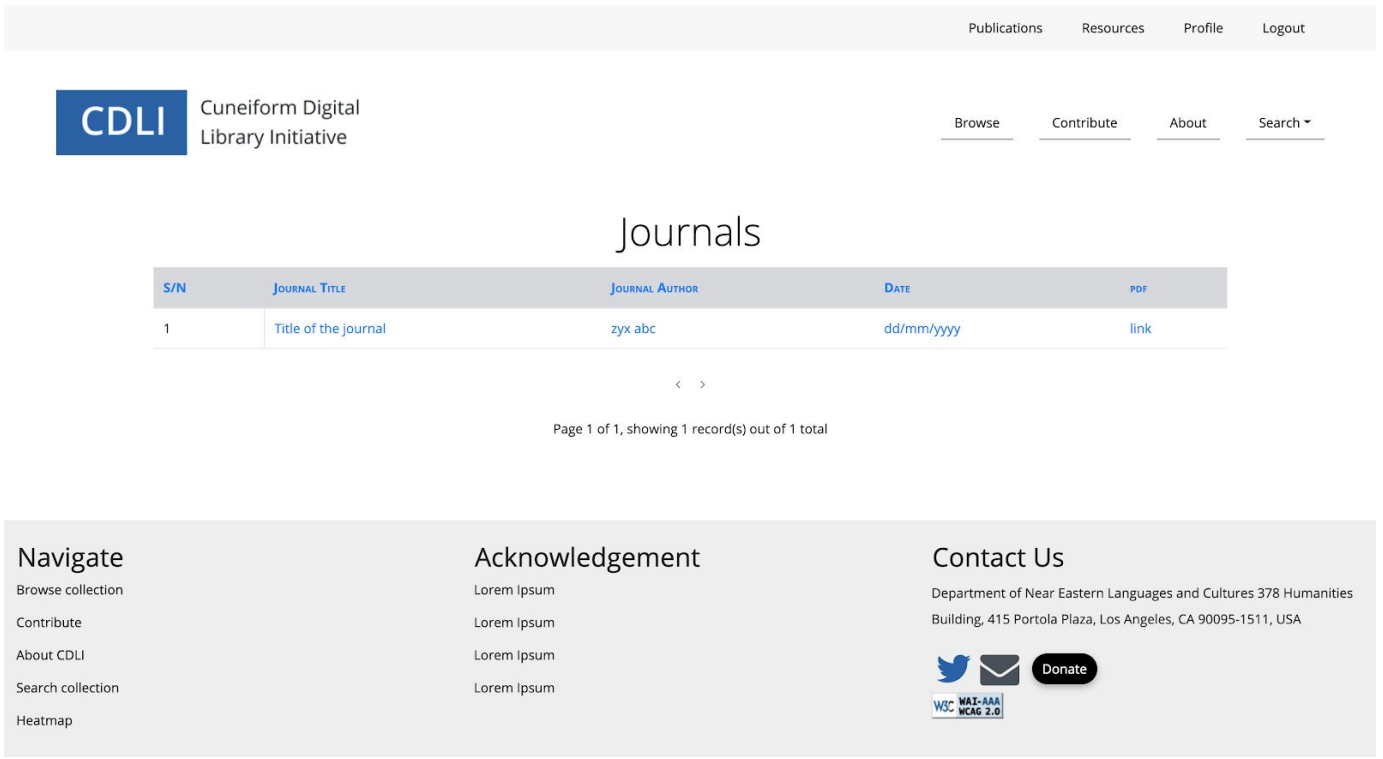
*(**Fig 7d :** when preview/edit journal button is clicked, modal with CKEditor to edit journal HTMLSourcecode)*



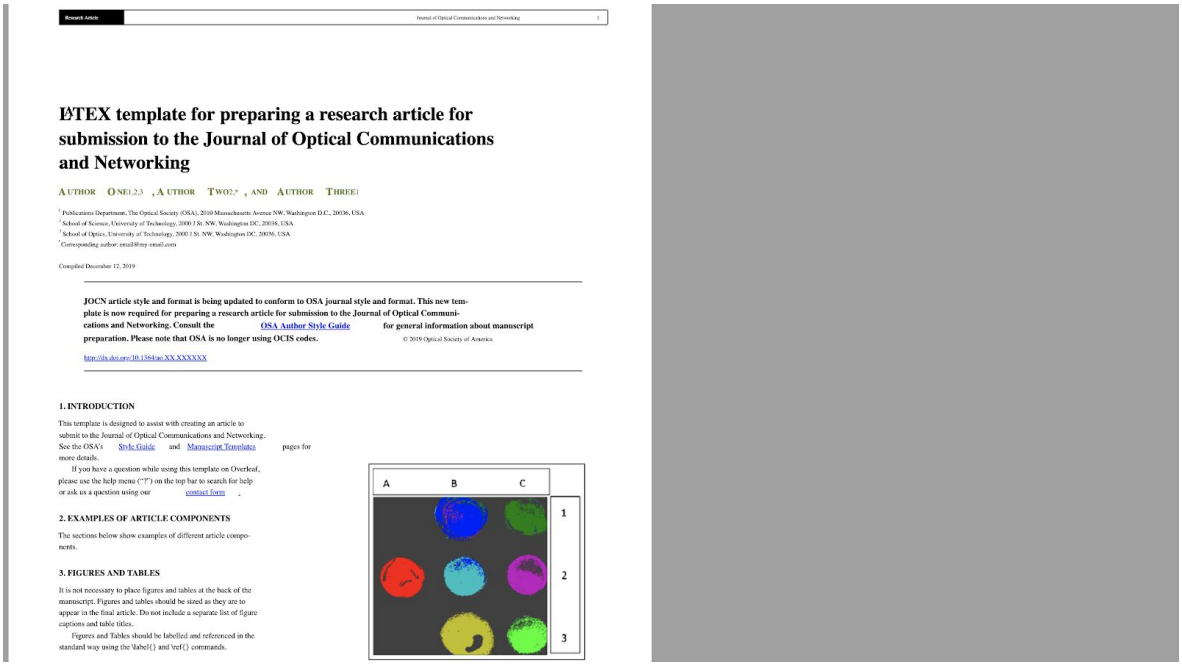
*(**Fig 7e :** When Journal is verified enable of submit button and show confirmation message)*



*(**Fig 7f :** Menu tab for the display of all types of journals)*



*(**Fig 7g :** View of CDLJ journal list)*



*(**Fig 7h :** sample view of CDLJ journal HTML)*

- https://book.cakephp.org/4/en/index.html
- https://www.onworks.net/programs/htlatex-online
- https://github.com/overtrue/phplint
- https://phpunit.de/documentation.html
- https://github.com/cdli-gh/Framework/wiki/Google-Summer-of-Code-2020-Ideas-List
- https://docs.docker.com/