



Request for Proposals (RFP)

LACI TIG25

LACI Support and Drupal Development

Atlanta Legal Aid Society, Inc.
54 Ellis St. NE
Atlanta, GA, 30303

Contact:
Virginia O'Reilly
vmoreilly@atlantalegalaid.org

Introduction and Purpose	2
Background	2
About Atlanta Legal Aid Society	2
LACI Overview	2
Scope of Work	3
Timeline	3
Procurement Phase	3
Project Phases	4
Evaluation Criteria	9
Proposal Submission Instructions	10
Confidentiality	13
Freedom of Information Act	13
Questions	13
Deadline	13

Introduction and Purpose

Atlanta Legal Aid Society (ALAS) seeks proposals from qualified bidders to help us adopt and extend LACI (Legal Aid Content Intelligence) as described below. We will use LACI to ensure that our publicly accessible self-help content, generally available at [Georgia Legal Aid](#), is kept up-to-date by alerting our document authors (known as “editors” and “reviewers”) when relevant online resources are updated. When the online resources change, we want to know as soon as possible so we can update our own information resources and self-help tools in response.

This RFP outlines the project background, the specific scope of Drupal development services requested, expected deliverables, and evaluation criteria. We strongly encourage applications from vendors with demonstrated experience in Drupal architecture, agile web development, or complex digital platform management within the cleantech, sustainability, or LACI-adjacent innovation sectors.

Background

About Atlanta Legal Aid Society

Founded in 1924, Atlanta Legal Aid Society (ALAS) provides free civil legal services to low-income individuals across the Atlanta metropolitan area. With over 100 years of service, our mission is to provide free civil legal assistance to help people in need access justice to protect their rights and secure their safety and stability. In recent years, ALAS has become a leader in legal technology, leveraging resources like Georgia’s LawHelp website, GeorgiaLegalAid.org, and LAVA (Legal Aid Virtual Assistant) to improve the accuracy and efficiency of legal information delivery.

LACI Overview

As legal professionals, we generate a lot of documents that explain the law to various people. These documents are memoranda to other lawyers, both in our firm and outside it. They are letters explaining legal matters to our clients. They are filings in court that explain the law to judges and other parties. And they are legal information that we put up on our website or hand out to the public.

Of course, the law we explain to others is not under our control. Instead, these “Authorities” are laws created by our legislatures, rules adopted by the courts, agency regulations, online forms, and other authoritative explanatory web pages. We need to know when these Authorities change, so we can make sure our explanatory documents are kept up to date.

To keep all of these documents up to date, we are employing LACI, the Legal Aid Content Intelligence software system. This software tracks all our relevant legal documents and allows us to register which external Authorities each document depends on. Georgia authorities, as well as US/federal authorities, are all available online, which means that LACI can check them periodically, and alert our reviewers and editors when a document needs to be reviewed. When the documents themselves change, reviewers and editors are notified to review to make sure LACI continues to have the correct Authorities.

You can learn more about LACI concepts and terminology from the information videos published on the LACI [YouTube channel](#). The LACI source code is available from our [Git repository](#).

This software is currently in use and has been active since May 2024.

Scope of Work

ALAS received new funding from the Legal Services Corporation (LSC) to adapt LACI for its use by deploying its own LACI servers, connecting LACI to monitor its own documents, and extending LACI to understand Georgia sources of law. ALAS is looking for a software development partner (“Respondent”) who is a Drupal module development expert and who can help us with at least the following LACI-related tasks:

- Developing the tools and infrastructure to allow ALAS to deploy an appropriately configured LACI server, from a git repository much like [the one used by Lone Star Legal Aid](#), in a testing and in a production environment.
- Developing two Drupal modules that will periodically check a web server for new or updated pages and update LACI accordingly.
- Developing a Drupal module to allow a LACI server to access online Georgia legal authority information: statutes, regulations, and court rules.

All code must conform to the [Coding Standards](#), with the exception that the copyright license on **new** files must refer to Atlanta Legal Aid Society as the copyright holder.

Timeline

Procurement Phase

- **RFP Issued:** April 20, 2026
- **Vendor Questions Due:** April 30, 2026
- **Responses to Questions Provided:** May 11, 2026
- **Proposal Submission Deadline:** May 29, 2026
- **Vendor Interviews (if applicable):** June 8 – 13, 2026
- **Contract Negotiation and Execution:** June 23, 2026

Project Phases

Based on experience we expect the effort to have substantially the following phases. Please estimate the percentage of total effort for each phase in your response and discuss how you would approach implementation in each phase.

1. **Discovery Phase.**

During the initial phase of the project, meet with ALAS team to discuss various initial requirements and to draft a proposed project timeline and communication plan, as well as:

- Location(s) of development, testing, and production servers.

- Confirmation of development methodology, use of git and branches, etc.
- Documentation of any items ALAS will need to have in place prior to development; e.g., DNS configuration, SMTP services, etc.
- Design wireframes, as appropriate, for any user-interactive elements of the project.
- Scheduling periodic follow-up meetings.
- Any other kick-off activities valuable to starting the project.

This phase will likely take two to four weeks.

Acceptance criteria: Deliver a project timeline, schedule meetings, and identify the other necessary project elements above.

2. Development Phases.

The following are the three deliverables:

a. Installing and configuring testing and production LACI servers.

ALAS needs to have the ability to install and configure LACI servers, at least one for testing and one for production use. Each should have a dedicated DNS name and use HTTPS for communication.

The servers should be hosted at a reliable provider who either provides Drupal-specific services or who provides virtual machines into which Drupal and its dependencies must be installed before LACI. The server must be able to send outbound email (using a third party service such as Sendgrid, Mailgun, etc.), must allow ALAS to execute “drush” commands to obtain backup files, and must allow ALAS to configure “cron” or another scheduling service to execute “drush cron” after hours when needed.

The deployment process must use Composer and must start from a git repository containing at least a composer.json file. Additional ALAS-specific code, configuration, recipes, etc. may also be in the repository. See an example repository [used by Lone Star Legal Aid](#) to deploy their LACI servers.

The process should have at least the following broad phases:

- Automated deployment of a Drupal server with the relevant LACI modules and theme installed.
- Automated or manual configuration of relevant settings and secrets; for example, API keys, SMTP configuration, etc.
- Automated or manual “**drush laci-restore**” to restore a backup of LACI information. It is neither necessary nor desirable to use a SQL dump as a backup/restore.
- All manual steps must be documented, and automated if possible.

ALAS will need adequate documentation and training to be able to accomplish these tasks using their own staff.

Acceptance criteria: ALAS staff must be able to use the tools and documentation developed in this phase to deploy an appropriately configured testing server and restore a LACI backup into that server. ALAS staff must be able to use the same tools and documentation to be able to obtain a backup of a running production server, to deploy an appropriately configured new production server, and to restore a LACI backup into that server.

b. **Developing two Drupal modules to add monitored Documents to LACI.**

LACI needs to know information about the legal documents the site is maintaining. It learns this information by accessing a cloud storage site and asking for an inventory of documents, including a unique identifier, a last modified date, an optional title, and an optional description. This information is used to create, update, or unpublish Document nodes in the LACI system.

Respondent will create two Drupal LACI modules to do this; each will access a different cloud storage source. These are:

- A URL to a sitemap (a page with either a **sitemapindex** of sitemaps, or a **urlset** of references). If the module retrieves a **sitemapindex** then it must iterate over the constituent **sitemap** entries. Each **url** is mapped to a LACI Document.
- A URL to the API endpoint of a site exposing the ScaleJustice API, accessing **resources** within such a system. Each is mapped to a LACI Document.

Each module will implement its own **DocumentSourcePlugin**. LACI will iterate over the configured plugins to update its list of LACI Documents. Each plugin will make an admin configuration page available which will expose at least these common elements (provided by **laci_core** and potentially augmented by the plugin):

- **enabled** (bool): “quick switch” to turn plugin operation on and off. If it is off, the common plugin code will log a message to the Drupal log messages warning that it is turned off.
- **minHoursBetweenChecks** (int): the minimum number of hours between times the plugin will access the cloud storage for information. Best practice is that the site’s “cron” will run at least hourly.
- **arrayOfSources** (array): an array of at least the following information:
 - **siteSrc** (text): the URL or other identifier for this cloud storage source.
 - **project** (term) (possibly optional): the default Project to assign to new Documents created from this source.
 - **arrayOfMonitors** (User) (possibly optional): the default Monitors (user accounts) to assign to new Documents created from this source.

The **arrayOfSources** allows a plugin to monitor *multiple* cloud storage sources. For example, different folders may belong to different Projects and have different sets of Monitors, or a LACI server might need to monitor two or more sitemaps as sources.

The plugin can allow empty fields for **project** and **arrayOfMonitors** if it can deduce this information from the data returned by the cloud storage system.

Each plugin will extend **DocumentSourceBase**, which will provide common methods to determine if it's time to access the cloud storage system (based on **enabled** and **minHoursBetweenChecks**), to build and process the common elements on the configuration form, and to create, update, and unpublish Documents based on the data retrieved by the plugin from its source. Respondent must complete the common elements in **laci_core** that support these plugins, including making the user experience of configuring the plugin more appealing and reliable. These common elements are currently committed to the repository on a feature branch.

To facilitate the ability to unpublish LACI Documents when an item is no longer present in the cloud storage, the plugin will need to associate each Document with a **sourcePlugin** (this module) and a **sourceID** (an identifier set by the plugin to uniquely identify each source; **can** be the same as the **siteSrc** parameter). This will be accomplished either in one of the ways below, or you may propose a different way:

1. Use a KeyValue Store collection for each module. This is easy to set up and use, and probably safe for up to a few thousand entries. *This is the current implementation.*
2. Create custom DB tables. This should scale indefinitely but is more complex.

LACI Documents created by a **DocumentSource** plugin will set these appropriately using a method on **DocumentSourceBase**, which will also provide methods to retrieve a set of LACI Documents that are associated with a (**sourcePlugin**, **sourceID**) pair. **The backup and restore functionality under \Drupal\laci_core\Drush\Commands must be updated to record and restore this information.**

On each check for Documents, each plugin will be checked to see if it's time to access the cloud storage system. If it is, for each **sourceID** associated with this plugin:

- For each item detected by the plugin from the cloud storage system:
 - If this item does not already have a LACI Document associated with it, the plugin must create a new LACI Document with the appropriate **project**, **link**, **title**, **description**, **monitors**, and document the relevant **sourcePlugin**, and **sourceID** for this

Document consistent with the implementation choice made for these data. A method to do this will be provided.

- If this item does already have a LACI Document associated with it, the plugin must update the “last modified date” of the LACI Document. A method to do so is available.
- For each LACI Document associated with this **sourcePlugin** and **sourceID** :
 - If there is no longer an entry in the relevant cloud storage system, unpublish this LACI Document and remove the relevant key\ValueStore entry.

Acceptance criteria: A LACI server with the two Drupal LACI modules delivered in this phase must be able to detect all existing web pages using sitemap.xml or using the ScaleJustice API as appropriate; add a new Document to LACI when there isn't one; unpublish a Document in LACI when the web page disappears; and update the “last modified date” of the related LACI Document when the web page is updated. The modules must be configurable using the admin pages described above. Common elements in **laci_core**, such as the config/setting form and the backup/restore commands, must work as described above.

c. **Developing a Drupal module to allow LACI to access Georgia legal authorities.**

LACI uses [Authsource Plugins](#) to access structured online authoritative information. Each structured source has its own plugin. The module that implements access for all the sources in a US State will typically have three or more plugins: statutes, regulations, and court rules. See [Federal](#), [Texas](#), and [Idaho](#) for examples.

Respondent must deliver a module that has essentially the same structure as the [laci-us-tx](#) and [laci-us-id](#) modules, including a taxonomy, and code for install, update (if necessary), and uninstall hooks and services. The taxonomy could be installed using a Drupal recipe. There is a repository already for [laci-us-ga](#) for this deliverable; deliver updates as pull-requests against this code. The module must retrieve online information of the following types from the following sources, and must allow Authorities to refer to sections (O.C.G.A. § 7-2-1) as well as sub-sections to two levels (e.g., O.C.G.A. § 7-2-1(a)(1)).

Type	Source	Example Citation
Georgia statutes	Justia https://law.justia.com/codes/georgia/	O.C.G.A. § 24-1-1
Rules of Civil Procedure	(see above)	O.C.G.A. § 9-11-1
Rules of Evidence	(see above)	O.C.G.A. § 24-1-1

Georgia regulations	https://rules.sos.state.ga.us/	Ga. Comp. R. & Regs. r. 125-2-3.04
Rules for: <ul style="list-style-type: none"> • Supreme Court of Georgia • Georgia Court of Appeals • Superior, State, Juvenile, Magistrate, Municipal, and Probate court. 	https://www.gasupreme.us/other-court-rules/	Ga. Sup. Ct. R. 1 Ga. Ct. App. R. 3 Unif. Super. Ct. R. 5 Unif. State Ct. R. [rule number] Unif. Juv. Ct. R. [rule number] Unif. Mag. Ct. R. [rule number] Unif. Mun. Ct. R. [rule number] Unif. Prob. Ct. R. [rule number]

Acceptance criteria: A LACI server with the module delivered in this phase must be able to access all Georgia statutes, regulations, and court rules from the sources described above.

3. Finalization Phase.

Fix any bugs and to bring software into compliance with the project specifications. This phase also includes any final required documentation and effort required to hand over the code, and any access and/or tools necessary to maintain the running environment.

We expect this last phase to take at least 10% of the total project time.

Acceptance criteria: All deliverables must work reliably in combination, running on both testing and production servers. ALAS staff are able to access all required source code and have demonstrated the ability to install and access a testing and a production server.

Evaluation Criteria

Each section below is assigned a specific weight for evaluation. Individual criteria within each section are considered collectively to determine the vendor's suitability for this Drupal development project and are not exclusive.

Relevant Experience and Qualifications (25%)

- Demonstrated expertise in Drupal architecture, custom module development, and theme implementation.
- Proven track record of delivering web solutions for cleantech, sustainability, or nonprofit innovation hubs like LACI.

- Familiarity with Drupal community standards, security best practices, and integration with third-party APIs.

Approach and Methodology (25%)

- Clarity, feasibility, and completeness of the proposed agile development lifecycle and project management approach.
- Ability to implement high-performance technical solutions while maintaining an intuitive user experience for diverse stakeholders.
- Alignment of the development methodology with the iterative, stakeholder-driven requirements of the LACI ecosystem.

Deliverables and Technical Quality (20%)

- Quality and robustness of proposed deliverables, including clean code, responsive design, and comprehensive technical documentation.
- Commitment to quality assurance (QA), automated testing, and a seamless deployment strategy.
- Provision for knowledge transfer and training to ensure LACI staff can maintain the platform post-launch.

Cost Proposal (20%)

- Overall cost-effectiveness and clarity of the firm fixed-price (FFP) bid in relation to the \$24,000 budget.
- Transparency regarding the allocation of resources, hourly rates (if applicable), and any potential licensing or hosting fees.
- Clear mapping of costs to specific project milestones and deliverables.

References and Past Performance (10%)

- Strength of references from comparable clients, specifically regarding Drupal-based web projects.
- Evidence of successful project outcomes, including meeting technical specifications and adhering to timelines in a professional services capacity.

Proposal Submission Instructions

Proposals must be twenty-five (25) pages or fewer, concise, well-organized, and submitted electronically in PDF format. Proposals must include the following components:

1. **Cover Letter:** Brief introduction of the firm, including an authorized point of contact.
2. **Respondent Information:** Company overview, history, client mix (percentage of nonprofit/government/commercial), and contact details (title, phone number, and email address).
3. **Experience and Qualifications:**
 - Describe your experience working with any non-profit organization for whom you provided a similar service.

- Specify the approximate percentage of business you received in the past year for performing similar evaluations.
 - List your personnel who will manage the services provided. This list must identify a point of contact who will manage the development process as well as a point of contact to manage business questions. Small organizations and individuals can name the same person in both roles.
 - Describe your proposed project and team organization. Identify key employees and/or supervisors.
 - List the certifications and credentials and experience of staff members, contractors, and subcontractors who would perform the work.
 - Provide a statement on whether the Respondent or any employee of the Respondent is related by blood or marriage to an ALAS employee or resides with an ALAS employee. If there are such relationships, list the names and relationships of said parties. Include the position and responsibilities within the Respondent's organization of such Respondent employees.
 - Describe your ability to meet scope requirements Section A – Evaluation Scope and B – Communication.
 - Provide samples or descriptions (links would be great if you can provide them) of your work on other similar projects.
4. **Cost Proposal:** Please submit a single firm fixed price (FFP) bid. Payment will be made upon the completion of each phase. The budget for this scope of work is \$24,000. Pricing must include all overhead costs.
 5. **Subcontractors:** If applicable, state the percentage of work performed by subcontractors and their qualifications.
 6. **References:** At least two (2) client references for similar work performed.

Atlanta Legal Aid's Rights

ALAS reserves the right to:

- Accept or reject any or all responses, or any part thereof.
- Waive any informalities or technicalities contained in any response received.
- Conduct discussions with Respondents and accept revisions of proposals after the closing date.
- Make an award based upon various selection criteria.
- Request clarification from any Respondents on any or all aspects of its proposals.
- Cancel or re-issue this RFP at any time.
- Retain all proposals submitted in response to this RFP.
- Invite some, all, or none of the Respondents for interviews, demonstrations, presentations, and further discussion.

Confidentiality

During the selection and project execution phases, ALAS may give you access to ALAS's confidential or proprietary information. You agree not to use this information for your or any

third-party's benefit and will not disclose this information to any person who does not have a need to know.

ALAS will not under any circumstances disclose any information submitted by Respondent to any other Respondents, except the questions and answers described above. ALAS will not disclose any information submitted by Respondent to ALAS to any other parties until after the contract is finalized.

Freedom of Information Act

The Freedom of Information Act (FOIA) and associated federal regulations may require ALAS to disclose certain documents to the public, including portions of your proposal. Generally, ALAS will not release any documents that would cause competitive harm to a Respondent or potential Respondent.

You are encouraged to label any confidential information contained in your proposal to facilitate ALAS's ability to withhold it from disclosure.

Questions

Submit in writing by May 4 2026, to Virginia O'Reilly, vmoreilly@atlantalegalaid.org. Please include "RFP for LACI TIG25 - Drupal Development" in the subject line.

Deadline

Proposals must be received by **5:00 PM (US/Eastern) on May 29, 2026**.