Documentation on how to use the application is provided in the How-To files. The PDF document is text only and the HTML document has animated images to illustrate the application user flow.

To access the code base that the application is built from:

- View the project repository on Github by navigating to https://github.com/UMN-Press
- The repository settings can be configured on the GitHub Organization account by selecting Settings at the right side of the repository options.

To invite users to access the repository:

- Select People from the repository options.
- Select 'Add a Member' and specify the email address, full name, or username that corresponds to the intended GitHub account.

To begin application Installation:

- Clone or Download the repository from Github by copying the Clone link or selecting the Download Zip button. Either unzip the archive or clone to the local machine by performing `git clone <repo-url>` (Replace <repo-url> with the copied URL from Github).
- Navigate inside the project folder and locate a file titled `database.sql`
- Using PostgreSQL, create a new database titled umn_app. If you wish to use a different name, be sure to change the database name inside `/server/modules/pool.js` on line 34.

- For more details, see the project README located within the root directory of this project.

To deploy the application using Heroku:

- Heroku is a Cloud Application Platform that will allow you to publish your apps to the web. It is free to use for development use, but offers paid tiers that offer more services.
- To deploy the app to Heroku, it must first have a Github repository (see above).
  - Sign up for an account on Heroku.com
  - Install Heroku CLI by typing `brew tap heroku/brew && brew install heroku` in Terminal
    - Additional installation notes and troubleshooting
  - Authenticate by typing `heroku login` in Terminal

Before you deploy, make sure your server `port` is configured correctly as:

```
const port = process.env.PORT || 5000;
```

Run the following commands from within your project folder.

- In terminal, navigate to your project folder and type `heroku create`
- Login in if prompted
- Type `git remote -v` to ensure it added successfully
- In terminal, type `git push heroku master`
- Our website is now live! However... we also have a database

Adding a PostgreSQL database on Heroku:

- In terminal, type `heroku addons:create heroku-postgresql:hobby-dev` to set up Postgresql on your Heroku project
- Next, type `heroku pg:push your_database DATABASE_URL` to copy your database contents up to Heroku. `your_database` is the actual name of your database (e.g. `umn_app`). `DATABASE_URL` is a heroku config variable created by the Add On. *Do not replace it with something else*, just type: `DATABASE_URL`. For example, if you were deploying the `umn_app` database, you should type `heroku pg:push umn_app DATABASE_URL`
- Update or create a module for your pg-pool configuration to the following code that will convert the heroku `DATABASE_URL` into a pool config object. The only line you should have to change is `database: process.env.DATABASE_NAME || 'umn_app'`. Change `your_database` to the actual name of your database. (e.g. `database: process.env.DATABASE_NAME || 'umn_app':`

To set up automated searches (no user interaction):

- Searches can be automatically scheduled by configuring the schedule.js module in the server directory.
- The module uses node-cron and the cron time string format to determine when the searches take place.
- By default, the server will run 6 batches of 450 searches at 1:00, 1:20, 1:40, 2:00, 2:20, and 2:40 AM server time.
- Note that the server must be running for the search to take place.

Known issues:

- Publication Items cannot be deleted from the database, be sure to review the CSV upload before accepting.
- When a spreadsheet upload is accepted and the page is redirected to the Publications tab, the user must refresh the page (**Mac**: ⌘ + R **Win**: Ctrl + R or F5) to load the Publication data on the webpage. Subsequent visits do not require this step.
- The buttons to cycle pages do not recognize the end of a list and will allow the user to continue clicking.