How to install software

- If someone wanted to deploy your application on their own environment, what should they do?

- What software is needed? (docker really shines here)

- What are sources of

- What external resources are used (put any free/paid tier information here)

The project can be cloned from the GitHub repository: <https://github.com/mslalor/Book-Library-Project>. It is not necessary to have .NET 6 or 7 downloaded onto your environment since the API is hosted, but that is an option. This project utilizes .NET 6, but .NET 7 is also compatible with this version. To run this project with .NET, ensure you are in the api folder of the project (cd api) and then run the command “dotnet watch run” in your terminal. Once the SwaggerAPI page is loaded onto your browser, the project is ready to run. The html file that holds our welcome page is called welcome.html in the main repository. However, the website is also located on <https://mslalor.github.io/Book-Library-Project/> and <https://personal-book-library.netlify.app/>.

This project utilizes Heroku, JawsDB, MySQL, Azure, SwaggerAPI, OpenLibrary, SendGrid, Netlify, and GitHub Pages. All of these resources are free and at most only require an account. The account with Heroku allows you to create an app with an add-on for JawsDB, which creates a free MySQL database for the user. This is the database used for this project since it is not a local database and can be accessed from our website. Heroku is also what connected the SwaggerAPI to allow for the connection of frontend and backend with .NET. OpenLibrary is the API used to populate the books users can search for and connects to the expansive book database. SendGrid is the application to send the reminder emails to renters. The API is hosted using the free tier of Azure, where it is published on an app service to allow for non-local connection. Finally, Netlify and GitHub pages both host the website without any fees.

How to use each completed feature

- List of features and how to use them (Should link to sprint features in the project)

- Document results of each function and how they link to other functions

Ex: "placed orders that have not been fulfilled can be viewed on the in process screen"..."there are three types of user accounts"

- How are external resources incorporated?

How to modify/extend software

Assuming someone has followed the instruction for installing the application, how can they make changes?

What compiler, languages? What build management? Where are the dependencies listed? Any automated builds?

Where is the backlog and the project bug lists? Call out major issue such as migrating to better resources or new major versions that require retrofitting

Should also communicate style expectations

How to run any existing automated testing. Location of test cases

FAQs

- Identify 3-4 frequency asked questions. This should focus on the end user.

- Identify any gotchas or problems you encourtered either in installing parts of the application

- Possible issues with external resources

https://about.readthedocs.com/?ref=readthedocs.com

https://docs.readthedocs.io/en/stable/tutorial/index.html --> https://about.readthedocs.com/pricing/#/community --> Best option as this free and provides enough resources for students to build their own docs and the can import their existing repo.