

Pride in Writing

Technical Report

Prepared By:

Pamela Vazquez

Gregory Raper

Rodrigo Estrella

Cliff Xu

Shyam Patel

OUR STORY

It takes a lot of courage to own up to who you are in a world where you are judged if you don't fit within the social norms. The world has been more open towards the LGBTQ community because the members of the LGBTQ community are so proud of who they are. We want people to continue to support the LGBTQ community and we hope that our website is a platform that showcases their amazing work and encourages people to deliberately support these writers.

Project Overview

This website aims to showcase work specifically written by members of the LGBTQ community to get people to buy their books, learn more about them as people and make them aware of which publishing houses are publishing minorities. This website will allow the user to navigate between books, authors, and publishers to see the relationship between them. The user will be able to filter books written by LGBTQ authors by genre, page count, and price; and they will be able to sort the books by year published and by its Goodreads book rating. If the user wants to start their search by author, they can sort the list of LGBTQ authors by year born, nationality, number of books written, and name; they can also filter by gender. If the user wants to search by Publisher, the user will be able to filter by country of origin, publication type, and will be able to sort by the number of authors showcased on our site, publisher name, and year the company was founded. We hope to make this website a place where people will learn about LGBTQ authors, buy books specifically written by LGBTQ writers, and see which publishing houses are supporting LGBTQ authors.

User Stories Received Phase 1

1. Author Achievements:

When I start to read a book, I would first look up the author. If this author has some awards or other achievements, for instance, one of his/her books sold over \$100 million, then I would consider this author's book worth reading. Maybe add this attribution in your Author model.

Response: Hi Yuhan, we are looking at presenting data related to the Author's awards. I am unsure if we will get around to this in phase 2 because it is not part of our plan. But we will take a look and try to get it done by Phase 3!

2. Book Vendor Locations:

As someone interested in reading some of these books, I would like to know where to order these books. I'm not too sure what vendors have these books in stock. Would it be possible to list some possible vendors where I can order these books off of such as Amazon? This would be extremely helpful to me because I would like to read some of these as soon as possible!

Response: Hi Eric, This is a bit related to another user story, but we are looking into adding this for Phase 2. Right now we can pull pricing and availability data from the Goodreads API, but this may not be representative of the best deals on that given book.

3. Author Hometown/City:

I think it would be neat to be able to see the author's hometown or city they were born in. Seeing an author's year and country of origin is already good data but being able to see the city they were born in may be useful for those that pride themselves in following authors from their hometown.

Response: Hi Kelly, I agree, and we will be trying to add this as part of Phase 2! Please let me know if you have any additional comments.

4. Where can we purchase the book:

Hi all, great job and I like your idea. However, I think the website could be more convenient if it can give customers a link to an online shop or something that could help customers to get those books easier.

Response: Hello, As part of phase 2 right now we will be adding price, and if available we will be adding a location/link to purchase the book as well. Right now we could do this using the Goodreads API, but it would give Goodreads prices, rather than the best price available.

5. Target Readers:

I know the authors are from the LGBTQ+ group. However, they may write some books that are related to LGBTQ+ or other topics. If the book could provide target readers such as the age range or women/men, that would be better for me to filter out the books that I don't want to read.

Response:

Hi Yuhan, I am a bit confused by your request. Would you like us to add the age and gender of the author, or something else? If it is that ladder that is something we have planned to add as part of Phase 2!

6. Publisher

If the publisher is famous, it must publish some famous books. As a customer, I would like to see what other books does this publisher published before when I click the link of the publisher. If I have read the book that this publisher has been published, I am willing to read other books that the publisher published.

Response: Hi Yuhan, as part of Phase 2 we are adding the notable works that have been published by the given publishing company. We should have that functionality by Monday, October 25th. Please let me know if you have any additional feedback.

7. Short Introduction of the book

As a customer, when I click the link to the book, I would like to see the short introduction of this book. This would make me know the story that the book is writing about. I would save a lot of time before I read it.

Response: Hi Yuhan, As part of phase two we are adding a summary to our book model page. Does this address your issue?

User Stories we gave Phase 1

1. Hey Development team, I am a middle school student who is responsible for delivering morning announcements over the intercom. The start of my announcement involves reading daily statistics for cases in our area. It would be awesome if the data could be

updated at a minimum, every day. This way I can provide the most recent, accurate information to my classmates.

2. Hey Development team, I am a mother of three so I am usually pretty busy throughout the day. I try my hardest to keep up with recent news and updates on COVID to keep my family as safe as possible. Sometimes, however, it's hard to keep up with so much; so, I was wondering if this website could contain a feature that displays major news regarding covid, for example, new outbreaks in the area, government announcements, etc. This would be awesome because it would be a centralized location that can keep me informed and my family safe.
3. Hey Development team, I am a pediatric doctor who hears from a lot of patients that they have trouble finding testing locations. As a result, I would think it would be amazing if there was a feature that showed nearby testing locations for users based on the address they search. Available and reachable testing is very important in beating this virus because we can take proper precautions for infected individuals.
4. Hey Development team, I am a math teacher at my local high school. After seeing numbers all day, I think my students and I can both agree we don't enjoy seeing a list of statistics. I request a feature that visualizes the data and can provide better visualization of the severity of these stats. Whether this is through charts, graphs, etc, I think this feature would better deliver the goal of this website in informing the population about COVID data.
5. Hey Development team, I am an elderly man who is a high-risk individual. I think this website will be great for me to identify the general threat of my surrounding neighborhood. But, because I am older I have trouble navigating complicated websites. I would love it if there was a feature that could make it easy to generate a search for my address. This would help because it would be friendly for me to take advantage of the data you provide and stay safe.

User Stories Received Phase 2

1. As someone who is older, it is difficult for me to click on the links on the left of each table precisely. Would it be possible to make the entire row a link to the instance page rather than just the left column? It would be extremely helpful for navigating pages!

Status: The authors, publishers, and book links to their respective instance pages have been made bold to make it more visible to click on and you can now also click anywhere in the row/card. We will see if there is anything else we can do in the next phase!

2. Hi, I took a look at the current version of the website and it was great to see the book ratings. It may be beneficial to see the average author rating or something like that so that I can look at highly rated authors first.

Status: Our API does not rate authors, it only rates the books themselves. We might not be able to do this unless we find an API that gives us author ratings.

3. Would it be possible to make a timeline of the books written for authors? It'd be awesome to see books the author has written in chronological order.

Status: I think we could sort the books written in chronological order, we'll take a look at this in the next phase.

4. Would it be possible to see where the publisher is located in a Google map format? I'd like to see where my publishers are located.

Status: At the moment we are planning on linking the current location of the publisher, we will have to see if we can implement this by using the Google Maps API in the next phase.

5. It's difficult as a user to navigate pages currently. If I scroll down on a page and click on a link to another page, it will stay scrolled to the bottom of the page. Would it be possible to automatically transport me back to the top of pages after navigation?

Status: We will fix this in the coming phase.

User Stories we gave Phase 2

1. I am an elderly person trying to use your website. I have terrible eyesight as it is, and the blurriness of your pictures bothers my headaches even more. If you could find clear pictures to place in your cards by the end of phase 2, that would be awesome.
2. I am an elderly person trying to gain information from your website. My brain has admittedly slowed down over the years. It is hard to read the large numbers on the covid page for the number of cases and the number of deaths. If you could add commas so it's easier to read, that would be appreciated.

3. I am a middle aged man using the website. I like the convenience it provides, but I do have a request that may improve convenience. Currently, I have to click on the date to see the data visualization for a covid entry. Because I like to check every day, it would be nice if the most recent entry's data was on the entry page before having to click on the date. For the rest of the dates, clicking will suffice I am just concerned with the most recent entry.
4. I am an elderly woman who is very concerned about covid. As a result, I am always tracking where covid is growing and prevalent. It would be appreciated if you could add a message on the home page with the most impacted country and city from the most recent covid entry.
5. Hi Development team, I am a geography teacher at a high school. I want my students to learn about locations covid is impacting, and have recommended your website as a source. An addition to your Country page that I believe would be helpful to conceptualizing covid's impact is to have the total number of cases for each country added to the table on the country page.

User Stories Received Phase 3

1. Currently when I go to a book instance page, the page count is shown as a floating point. It doesn't seem normal for a book to contain part of a page. Would it be possible to turn it into an integer?

Status: Hey Eric,

Totally missed that, we will take a look and if we have time, we will turn it into an integer!

2. Currently the number of instances shown on the model page at a single time is a fixed number. Would it be possible to allow me to change the page size for pagination? This would allow me to see more instances at once in the model page.

Status: Hey Eric,

We are still working on cleaning up some stuff with pagination, therefore we will need to take a look at this in the next phase.

3. Currently when I click on an instance page and it is not loaded, it will display "test paragraph". This is really distracting me from browsing through the content. Would it be possible to change this to a loading animation? Please also remove the "test paragraph" on the home page.

Status: Hey Eric,

We will change this before the next phase!

4. It's currently difficult as a user to tell if there are no connections with other models on instance pages. Currently it just displays a blank space if there are no connections. Could you either add a loading animation or say no connections when there are no connections to other models?

Status: Hey Eric,

Thanks for the advice. We will definitely update this feature in this phase.

5. Currently book ratings are shown as a floating point. Would it be possible to show out of five stars the rating of the books? This would be quite helpful at identifying at a glance how highly rated a book is.

Status: Hi Eric,

Thanks for the feedback. This is an amazing feature addition that we appreciate, and will implement it in this phase.

User Stories we gave Phase 3

1. Hi Development Team,

I really enjoy your website! I am visually impaired so sometimes I find it hard to see where my mouse is. I would love if you could add a feature where the text color of a car.

2. Hi Development team. I am a fan of the website, in particular the countries page. I love the addition of flags. However, the zero frame between the top of the box and the flag is slightly displeasing. If this could be fixed that would be awesome.

3. Hey COVIDDB team, I am a fan of the website. On the cities page it seems a little illogical in the way the grid is listed. It would make more sense to me if country and population were listed before latitude and longitude. Thanks!

4. Hey Covid DB team. Awesome website, I love the information it provides to me and my family. I was wondering if countries could be listed in the grid alphabetically. This way it would make more sense and flow more logically. Thanks!

5. Hey COVIDDB team. I am a avid user of your website, and I am grateful for the information you provide. I have a request for cities to be arranged alphabetically by name. I think this would provide great logical flow to the website.

Restful API

<https://documenter.getpostman.com/view/17727241/UUy1f71j>

Authors API Endpoints

- **GET Authors**

<https://prideinwriting.me/api/authors>

Returns a list of all of the authors stored in this API with basic information on the author, such as year born, nationality, name, gender, biography, other genres of writing, Goodreads page, and lifespan.

- **GET Authors by author id**

<https://prideinwriting.me/api/authors/id=<id>>

Returns the given author for any author id, such as year born, nationality, name, gender, biography, other genres of writing, Goodreads page, and lifespan.

- **GET Authors by Author IDs**

<https://prideinwriting.me/api/authors/ids=<ids>>

Returns the given author's information for the respective author ids listed, such as year born, nationality, name, gender, biography, other genres of writing, Goodreads page, and lifespan.

Books API Endpoints

- **GET Books**

<https://prideinwriting.me/api/books>

Returns a list of all of the books stored in this API with basic information on the book, such as genre, year published, Goodreads rating, page count, price, Goodreads Reviews, summary, authors, e-book, and publishers.

- **GET Books by book id**

<https://prideinwriting.me/api/books/id=<id>>

Returns the given book for any book id stored in this API with basic information on the book, such as genre, year published, Goodreads rating, page count, price, Goodreads Reviews, summary, authors, e-book, and publishers.

- **GET Books by Book ID's**

<https://prideinwriting.me/api/books/ids=<ids>>

Returns the given books for the respective books ids listed with basic information on the book, such as genre, year published, Goodreads rating, page count, price, Goodreads Reviews, summary, authors, e-book, and publishers.

Publishers API Endpoints

- **GET Publishers**

<https://prideinwriting.me/api/publishers>

Returns a list of all of the publishers stored in this API with basic information on the publisher, such as country, publication type, authors in our website showcased, published authors, publisher name, year founded, parent company, location, contact us link, website link, and publisher history.

- **GET Publishers by publisher id**

<https://prideinwriting.me/api/publishers/id=<id>>

Returns the given publisher for any publisher id stored in this API with basic information on the publisher, such as country, publication type, authors in our website showcased, published authors, publisher name, year founded, parent company, location, contact us link, website link, and publisher history.

- GET Publishers by Publisher ID's

<https://prideinwriting.me/api/publishers/ids=<ids>>

Returns the given publisher for any publisher id listed with basic information on the publisher, such as country, publication type, authors in our website showcased, published authors, publisher name, year founded, parent company, location, contact us link, website link, and publisher history.

Models

Author

- Author Name - Sort
- Author on tour - Filter
- Author Summary
- Author image
- Year Born - Sort
- Nationality - Sort
- Genre - Sort
- Notable Works
- Books connections
- Publishers connections

Books

- Book Name - Sort
- Genre - Filter
- Publisher Name - Sort
- Year published - Sort
- Book rating - Sort
- Maturity rating - Sort
- Page count - Filter
- Price - Filter
- Summary
- Book Image
- Publisher connections
- Authors connections

Publisher

- Publisher Name - Sort
- Publisher Image

- Country of origin - Filter
 - Publication Types - Filter
 - Year company founded - Sort
 - Parent Company / Owner - Sort
 - Location of current office - Filter
 - Website link
 - Author Connections
 - Book connections
-

Filterable/Sortable Attributes:

Author

- Nationality - Sort
- Number of books written - Sort
- Name - Sort
- Gender - Filter
- Year Born - Sort

Books

- Year published - Sort
- Goodreads book rating - Sort
- Page count - Filter
- Price - Filter
- Genre - Filter

Publisher

- Publication Types - Filter
- How many authors showcased on our site have been published - Sort
- Publisher Name - Sort
- Year company founded - Sort
- Country of origin - Filter

Searchable Attributes:

The user can search by the title of the book, name of author, and name of the publisher.

Media

Author: Bio, the image of the author

Books: Summary/description, an image of the book cover
Publisher: Bio/History of Publisher, an image of the logo

Connections

Author: Connects to Books because the author writes books. It also connects to Publisher because, to be a published author, an author must work with a publishing company.

Book: Connects to Author because the books are written by authors and it also connects to Publisher because a book can be printed and published through a publishing company.

Publisher: Connects to Books because publishers print and publish books to make a profit out of them. Also connects to authors because, to have books to publish, the publishing company must work with the authors of the books.

TOOLS

Back-end

- AWSAmplify: used to deploy our web application
- Postman: used to design our Restful API
- NameCheap: used to register the Pride in Writing domain name
- Elastic Beanstalk: Used to host our backend API
- Postgres: Used to store the tables/data for our models

Front-end

- React: used to build our web app
- Material UI: used to create the user interface in our React application
- GitLab: used for our project repository and development across team members
- Selenium: Used to test GUI of our website
- Jest: Used to test React and Typescript components.

Restful APIs and Data Sources

- Restful APIs and Data Sources: used to scrape information that will be displayed in our web application
 - a. <https://developers.google.com/books/docs/overview>
 - i. We used the Google Books API to get our book information by making API requests like this one <https://www.googleapis.com/books/v1/volumes>
 - b. <https://openlibrary.org/developers/api>
 - i. We used the Open Library API to get our book and author information by making API requests.
 - c. https://en.wikipedia.org/wiki/List_of_LGBT_writers
 - i. Used to get general information of LGBT writers
 - ii. We made a scraper to scrape all of the authors and images of the LGBTQ Authors by scraping each table and grabbing the names of the authors and then scraping the "href" reference from the a tags and concatenated it to the string Wikipedia URL. This allowed us to make a request to the Wikipedia page of a certain author. Once inside the Wikipedia page of the author, we scraped the "infobox vcard" which is where the image of the author is located. We grabbed the link of the image that is in the "infobox vcard" and assigned wrote it to a CSV file mapped to the author. Finally, we looped until we ran out of authors and got the resultant images in a CSV file.https://en.wikipedia.org/wiki/List_of_English-language_book_publishing_companies
 - iii. Used to get general information of English-language publishing companies like pictures, and parent companies
 - iv. We made a scraper to scrape all of the publisher names, images, and information from the list of English publishers from the Wikipedia Link. We did it by scraping the lists of every letter of the alphabet and grabbing the names of the publishers and then scraped the "href" reference from the a tags and concatenated to the string "https://en.wikipedia.org". This allowed us to make a request to the Wikipedia page of a certain publisher. Once inside the Wikipedia page of the publisher, we scraped the "infobox vcard" which is where the image and the information of the publisher is located. We grabbed the link of the image that is in the "infobox vcard" plus all of the information in that "infobox vcard" and stored it in label and data. Once the loop was done for a certain publisher we output it to a CSV file and then the loop continued for all the other publishers.

HOSTING

The domain name was registered from NameCheap, we then used a Custom DNS to connect our NameCheap domain to the nameservers provided to us by AWS Amplify's Route53. We also used AWS Amplify to connect our app through GitLab. To deploy our app, we simply need to run yarn install and yarn start, and the website is deployed!

Phase 2 Features

Description of how you implemented your database and pagination.

Pagination:

The pagination for our website is currently being handled in the front-end. The view-all pages for our models make a call to the getModel endpoints to fill in the page with all of their respective data.

The Authors' and Publishers' pages were populated with their respective information and displayed in a table format that was implemented through the Material Table component from the Library. This component implements Pagination itself including functionality like navigating through pages by clicking the previous and next buttons and being able to see the total records per model.

The Books page was also populated with its respective information but it was displayed in a grid format that was implemented through Material UI pagination and its custom Card component. Therefore the pages were comprised of 9 cards each displaying an instance of the book data. Similar to the Material Table Pagination functionality, the Material UI pagination provides the ability to navigate through pages by clicking the previous and next buttons and also displays the total Book records.

Database:

In implementing the database, our team decided to create a PostgreSQL database on AWS RDS. In setting up the database, there was a series of steps we followed that

generated a database that we were able to access through pgAdmin. In pgAdmin, we created a server to access the database and set information such as port and endpoint. Our team created one database per model. Although there are a few overlapping elements such as the model image, we wanted to make a distinction between the unique elements of each model. Since there are only three models, we decided that the compartmentalization of databases provides an organization that outweighs the overhead of three databases. In populating the databases, we found APIs and data sources online that we would pull. There were separate scripts for each database that was responsible for scraping or obtaining data. Any data that was obtained was converted into a CSV file, where we leveraged a script to push the necessary fields to the database.

Phase 3 Features

Description of how you implemented sorting/searching/filtering

Pagination:

In the back-end:

Pagination is handled in the API. First we query the database for all the needed information for the certain model. Once we queried the database, if the perPage parameter is empty we return all the records that the query returned. Otherwise, if the perPage parameter is set we call the paginate() function to return the desired records per page. Afterwards we call the dump() function on the desired schema and change the query object into JSON. Finally we call the jsonify() function to format it and send it to the front end.

In the front-end:

Pagination is handles by various components including those from materialUI. There are two pagination systems one is for Authors/Publishers list pages, the other for the Book list page. The page is kept track of as a React component's state. This causes the page to update and reload data when the page is changed. Additionally the page is saved as a local storage variable to retain the same page upon refresh.

Sorting:

In the back-end:

We implemented sorting through the API by passing in a column and ordering the data by ascending or descending order based on what the query's returned from our database.

In the front-end:

We implemented sorting with the "sort" buttons. We used the states in order to store the user's selection value. After, we passed the selected value to our method that handles the API. Data returned is displayed on the table in the desired sort.

Filtering:

In the back-end:

The back-end implemented filtering by using the filter function from SQLAlchemy to filter by the object we received from our query.

In the front-end:

We created different text fields for each of the filters on the model pages. These text fields will trigger a search if the user hits enter. There are also select components used to allow user selection from a preset list. Upon selection the user should see the table be updated with new data from an API call with the given filtering query parameters.

Searching:

In the back-end:

We implemented the searching by utilizing SQLAlchemy and its filtering. We pass in a query that will be searched on but the query is split up and each of those words gets filtered on. We had to tailor lowercase and uppercase queries by parameter to ensure the searches worked with the data in our database tables.

In the front-end:

We created a highlighting function that highlights the corresponding search string. In the book card view however, we imported a highlighting component from `react-highlight-word`. We had to wrap all of the text on screen in these components to ensure they could be highlighted. In order to display their respective cards/rows we passed the search query to the back-end.

GitLab

<https://gitlab.com/JunLum/pride-in-writing>