IMilestone list

- **Make test database - MM/DD**
    - In order to test our front- and back-end code, we'll need to construct some sort of testing database that is relatively small in size, but still large enough to test the necessary components of our generator. Additionally, it will need to be formatted in some way that is efficient and preserves the book information we would like to make queries based on (length, rating, genre, etc.). A deliverable for this project could consist of a small snippet of our database in the form of a screenshot, where our data formatting is completely shown and easy to understand. Some labels may be used as necessary.
- **Make API database* - MM/DD**
    - Upon completion of the test database and the components that require it, we will expand our test database into something more comprehensive. To do this, we will likely need to use some sort of data scraping method to obtain a large amount of books. To accomplish this, we will look into the API of several major book-related services in order to determine the best means of gathering book information into a database. Similarly to the test database, we can provide a snippet of the data as our deliverable, possibly alongside a count of the books in the database to compare to the test database's size.
- **Make a front end - MM/DD**
    - In order to accommodate for user input, a front end for the application must be created. There has to be some sort of input for the user's query, and an output for the to-be-read booklist, and each aspect of this front end must be functional and intuitive. A potential deliverable for this milestone could be a screenshot of this front end, displaying all of the components necessary for this project.
- **Make a back end - MM/DD**
    - Once the input is obtained from the user, it must be read into a form understandable by the generator, then processed in conjunction with our database to create a to-be-read booklist. Deliverables for this area of our project could include records of successful postman messages, as well as code snippets or screenshots as necessary.
- **Push info from front end to back end - MM/DD**
    - Once the information is input by the user, it must be pushed from this front-end web interface to the necessary back-end code, which will require communication between these two distinct areas of our project. One possible deliverable for this could be screenshots of the relevant code, as well as test inputs/outputs to verify the functionality of the front-end to back-end push.
- **Push info from backend to front end - MM/DD**
    - The deliverable screenshot of data from database
    - After the generator is run and a list is created, it must be delivered to the front-end and formatted into a readable output for the user, requiring an additional amount of communication between the front- and back-ends. Once this is completed, potential deliverables could include screenshots of relevant data

from the database, code snippets, and visual demonstrations of the back-end to front-end push.

*Given adequate completion of other main milestones.

| Task | Person responsible | dates |
|---|---|---|
| Implement a database that holds book information. | Emma | Fri Oct 22, 2021 |
| Write basic CRUD operations on backend. | Emma | Fri Oct 29, 2021 |
| Query the database with the inputs from the user. | Emma | Fri Nov 5, 2021 |
| Redirect booklist data from the database back to the frontend. | Emma | Fri Nov 12, 2021 |
| Create a backend repository on github. | Emma | Fri Oct 15, 2021 |
| Design a web-based interface for taking in book-related query information from the user. | Nolan | Fri Oct 15, 2021 |
| Create a visually informative output for displaying a to-be-read booklist generated from the application. | Nolan | Fri Nov 5, 2021 |
| Research API of existing services to find ways to create an extensive database of books. | Nolan | Fri Nov 12, 2021 |
| Analyze existing book information to determine an informative and efficient means of storing books within a database. | Nolan | Fri Oct 29, 2021 |
| Refine information retrieval methods to find the best means of generating to-be-read booklists based on user input. | Nolan | Fri Oct 22, 2021 |

| task | % of effort Emma | % of effort Nolan |
|---|---|---|
| Implement a database that holds book information. | 100 | 0 |
| Write basic CRUD operations on backend. | 100 | 0 |
| Query the database with the inputs from the user. | 100 | 0 |
| Redirect booklist data from the database back to the frontend. | 100 | 0 |
| Create a backend repository on github. | 100 | 0 |
| Design a web-based interface for taking in book-related query information from the user. | 0 | 100 |
| Create a visually informative output for displaying a to-be-read booklist generated from the application. | 0 | 100 |
| Research API of existing services to find ways to create an extensive database of books. | 0 | 100 |
| Analyze existing book information to determine an informative and efficient means of storing books within a database. | 0 | 100 |
| Refine information retrieval methods to find the best means of generating to-be-read booklists based on user input. | 0 | 100 |