# GoodBooks: Book Reviews And Statistics

Gajera Dev(2016EE10431), Harshit Jain (2016EE10443), Sarvesh Nalawade (2016EE10442)

March 1, 2020

## 1 Introduction

Goodreads is a social cataloging website that allows individuals to freely search its database of books, annotations, and reviews. It is also the world's largest platform for book readers and authors to manage their bookshelves, share and review and recommend the books to their peers.

Inspired from the same, we have created a basic website where users can sign up/login to find the best rated books and authors, recently published books, etc. One can also search for a book information using anything among book title, author name, isbn number, publication year or tags. In addition, we have also implemented the functionality for advanced book search in which the user can search for a book using the combined info.

We envision our project as a service platform which allows the readers worldwide to know about their favorite authors, genres, etc. and at the same time manage their own bookshelf.

### 1.1 ER Diagram

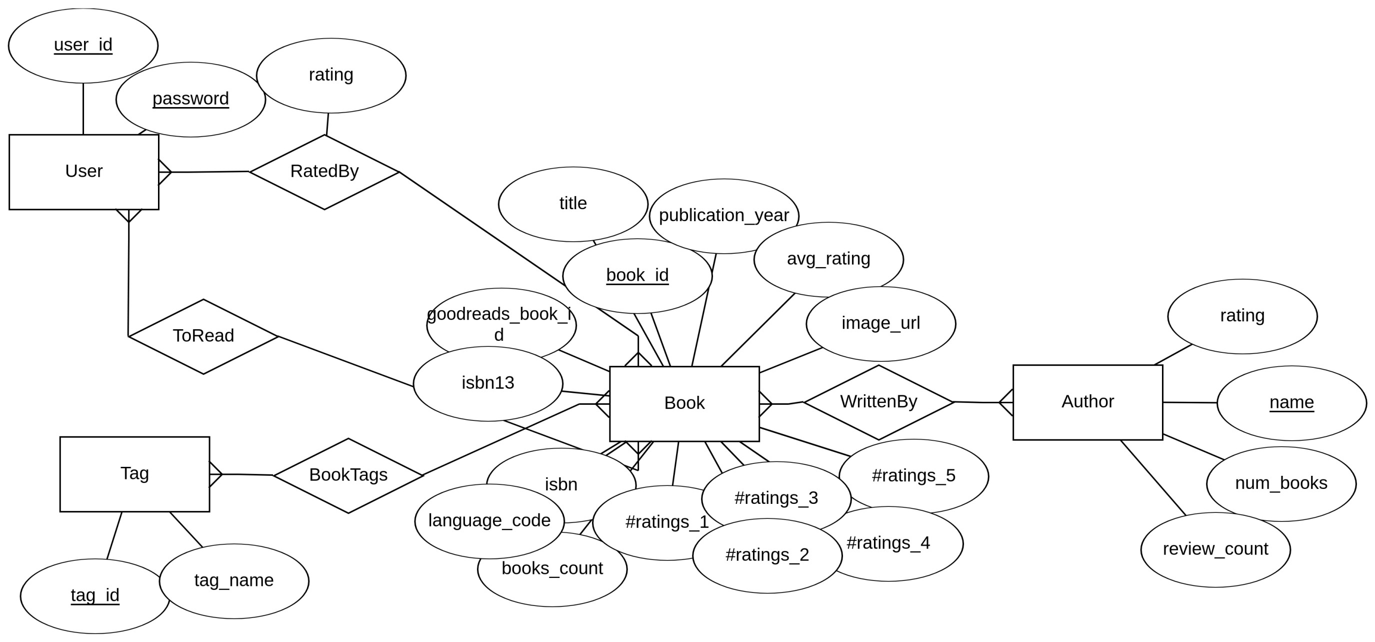


Figure 1: ER Diagram

### 1.2 Entities and attributes

| Entity | Attributes |
|---|---|
| Book | book_id, title, publication_year, language_code, isbn, isbn13, goodreads_books_id, books_count, avg_rating, #rating_1, #rating_2, #rating_3, #rating_4, #rating_5, image_url |
| Tag | tag_id, tag_name |
| Author | name, rating, num_books, review_count |
| User | user_id, password |

Table 1: Entities and Attributes

## 2 Data Description

- Data Source: https://github.com/zygmuntz/goodbooks-10k
  The above git repository contains: books.csv, ratings.csv, tags.csv, to_read.csv and book_tags.csv

| CSV File | Column names |
|---|---|
| books.csv | book_id, title, publication_year, language_code, isbn, isbn13, goodreads_books_id, books_count, avg_rating, #rating_1, #rating_2, #rating_3, #rating_4, #rating_5, image_url |
| ratings.csv | user_id, book_id, rating |
| tags.csv | tag_id, tag_name |
| book_tags.csv | book_id, tag_id |

Table 2: Structure of raw dataset

- Data processing and cleanup
  - Removed duplicate entries from `book_tags.csv` to enforce unique and referential constraints in the table.
  - Generated random strings as passwords for all the users using a python script and saved this in `users.csv` for authentication purposes.
- Since the number of tuples removed from the downloaded data are very less with respect to the total number of tuples, size of raw dataset and the dataset after cleaning is approximately the same. Hence, in Table 3, we have only included the final file sizes.

| Table Name | No of Tuples | Time to Load | Size of File |
|---|---|---|---|
| Books | 10000 | 211.033 ms | 3.3 MB |
| Users | 53424 | 258.271 ms | 897.1 KB |
| Ratings | 5976479 | 249950.195 ms | 72.1 MB |
| Tags | 34252 | 619.507 ms | 722.5 KB |
| BookTags | 999904 | 27284.667 ms | 15.7 MB |
| ToRead | 912705 | 25062.480 ms | 9.4 MB |

Table 3: Data Statistics

## 3 Database and Query Information

### 3.1 View of the System

- Sign up/Login page [Figure 2]:
  - User can sign in using already existing user id and password leading to user page.
  - In case of a new user, signup option assigns them a new user id and password.

  - The moderator can login to the system using user id and password as `admin`.

- User page [Figure 3]: Clicking on book title, author name leads to book, author page respectively
  - Log Out button: Leads to Signup/Login page.
  - Change Password button: Asks the user for a new password and updates it in the database.
  - Best Rated Books button: Shows the list of books with descending order of average rating.
  - Best Rated Authors: Shows the list of authors with descending order of rating.
  - Most Popular Books: Shows the list of books with descending order of rating count.
  - Most Popular Authors: Shows the list of authors with descending order of rating count.
  - Most Active Users: Shows the list of user ids with descending order of count of books reviewed.
  - Most Recent Books: Shows the list of books with descending order of publication year.
  - User Specific ToRead: Shows the list of books marked to read by the logged in user.
  - User Rated Book: Shows the list of books rated by the logged in user.
  - Search Queries: Search the book by Title or Author or Tag or Year or ISBN.
  - Advanced Book Search: Search the book by using partial title, author name, publication interval, tags or rating intervals in conjunction.

- Book page [Figure 4]:
  - Show Book's Cover page photo, title, author names, publication year, ISBN number, language code, average rating, number of ratings, rating_1, rating_2, rating_3, rating_4 and rating_5.
  - User Rating: Shows rating that the user has given to the book and in addition, also provides with the functionality to change the rating. Here, the rating table of the database is updated and a trigger updates the books table, which results in the new values of ratings and average rating.
  - To Read: Shows whether the user has marked the book as toread or not. Additionally, the toread status can be altered by clicking on Yes or No buttons as a result of an sql update query executed in the back end.

- Author page [Figure 5]:
  - Author Details: Show author's name, rating, number of books written and the total rating count.
  - Books Written: Show all the book titles and cover page images sorted by average book ratings. For each book, we also display the number of users who have marked the book as to-read.

- Moderator page [Figure 6]:
  - Add Book: A form to add a new book to the database by filling out the required details.
  - Add Tag: A form to add a new tag to any book with the given goodreads_book_id.

## 3.2 Database And Special Functionalities

### 3.2.1 Constraints

Constraints are the rules enforced on the data columns of a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database. Hence, for the same, we have included several constraints as mentioned in the table below.
Each table has been assigned a primary key considering the frequency of usage of different columns in the queries used in developing the portal functionality. In addition, there are `Foreign key constraints` to ensure referential integrity in the database and several `uniqueness and not null` constraints as well. Some Foreign key constraints have also been assigned the required deletion policies to ensure correctness across the tables while we serve the users.

| Table Name | Column Name | Constraint |
|---|---|---|
| Books | book_id | PRIMARY KEY |
| Books | good_reads_book_id | UNIQUE, NOT NULL |
| Books | authors | NOT NULL |
| Books | title | NOT NULL |
| Books | average_rating | NOT NULL, >=0, <=5 |
| Books | ratings_count | NOT NULL, >=0 |
| Books | ratings_# | NOT NULL, >=0 |
| Books | image_url | NOT NULL |
| Users | user_id | PRIMARY KEY |
| Users | password | NOT NULL |
| Tags | tag_id | PRIMARY KEY |
| Tags | tag_name | UNIQUE, NOT NULL |
| Ratings | user_id, book_id | PRIMARY KEY |
| Ratings | user_id | REFERENCES Users(user_id) |
| Ratings | book_id | REFERENCES Books(book_id) |
| Ratings | rating | >=0 , <=5 |
| BookTags | goodreads_book_id, tag_id | PRIMARY KEY |
| BookTags | goodreads_book_id | REFERENCES Books(goodreads_book_id) |
| BookTags | tag_id | REFERENCES Tags(tag_id) |
| BookTags | count | ==-1 or >=0 |
| ToRead | user_id, book_id | PRIMARY KEY |
| ToRead | user_id | REFERENCES Users(user_id) |
| ToRead | book_id | REFERENCES Books(book_id) |

Table 4: Constraints table

### 3.2.2 Indexes

An index is a schema object used by the server to speed up the retrieval of rows by using a pointer. It can reduce disk I/O(input/output) by using a rapid path access method to locate data quickly. Furthermore, it should be kept in mind that although, it helps to speed up select queries and where clauses, it slows down data input, with the update and the insert statements.
Since, we have a lot of queries that use book title to search the book details from the database, we have added an index on title in the Books table. In addition to this, the postgresql also defines indexing on the primary keys which further aids our query execution.

```sql
CREATE INDEX Books_title_index ON Books(title);
```

### 3.2.3 Views

In SQL, a view is a virtual table based on the result-set of an SQL statement and materialized views are disc-stored views which are defined by a database query like normal views however, their underlying query is not executed every time you access them. These give us the functionality to create custom views of the dataset which are more easy to use and execute the queries faster.
In accordance with this, we have created a materialized view Authors, for storing information about the authors. This view includes author name, number of books written, their rating and the number of reviews.

```sql
CREATE MATERIALIZED VIEW Authors AS
    SELECT ath AS author, sum(average_rating*ratings_count)/sum(ratings_count) AS
    rating, count(book_id) AS num_books, sum(ratings_count) AS review_count
    FROM (SELECT *, regexp_split_to_table(authors, ', ') AS ath FROM books) AS t1
    GROUP BY ath
    ORDER BY review_count DESC, rating DESC;
```

### 3.2.4 Triggers

A Trigger is a stored procedure in database which automatically gets invoked whenever a special event in the database occurs. Triggers can be written for generating some derived column values automatically, enforcing referential integrity, etc. In our application, we have defined triggers to update the average_rating, #ratings whenever some user adds, removes or updates the rating given to any book.

```sql
CREATE OR REPLACE FUNCTION insert()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Books
    SET ratings_count = ratings_count + 1,
    ratings_1 = CASE WHEN NEW.rating = 1 THEN ratings_1 + 1 ELSE ratings_1 END,
    ratings_2 = CASE WHEN NEW.rating = 2 THEN ratings_2 + 1 ELSE ratings_2 END,
    ratings_3 = CASE WHEN NEW.rating = 3 THEN ratings_3 + 1 ELSE ratings_3 END,
    ratings_4 = CASE WHEN NEW.rating = 4 THEN ratings_4 + 1 ELSE ratings_4 END,
    ratings_5 = CASE WHEN NEW.rating = 5 THEN ratings_5 + 1 ELSE ratings_5 END,
    average_rating = (average_rating*(ratings_count - 1) + NEW.rating)/ratings_count
    WHERE book_id = NEW.book_id;
    RETURN NEW;
END; $$ LANGUAGE 'plpgsql';

CREATE TRIGGER RatingInsert
    AFTER INSERT ON Ratings FOR EACH ROW EXECUTE FUNCTION insert();
```

```sql
CREATE OR REPLACE FUNCTION delete()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Books
    SET ratings_count = ratings_count - 1,
    ratings_1 = CASE WHEN OLD.rating = 1 THEN ratings_1 - 1 ELSE ratings_1 END,
    ratings_2 = CASE WHEN OLD.rating = 2 THEN ratings_2 - 1 ELSE ratings_2 END,
    ratings_3 = CASE WHEN OLD.rating = 3 THEN ratings_3 - 1 ELSE ratings_3 END,
    ratings_4 = CASE WHEN OLD.rating = 4 THEN ratings_4 - 1 ELSE ratings_4 END,
    ratings_5 = CASE WHEN OLD.rating = 5 THEN ratings_5 - 1 ELSE ratings_5 END,
    average_rating = (average_rating*(ratings_count + 1) - OLD.rating)/ratings_count
    WHERE book_id = OLD.book_id;
    RETURN NEW;
END; $$ LANGUAGE 'plpgsql';

CREATE TRIGGER RatingDelete
    BEFORE DELETE ON Ratings FOR EACH ROW EXECUTE FUNCTION delete();
```

```sql
CREATE OR REPLACE FUNCTION update()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Books
    SET ratings_1 = CASE WHEN NEW.rating = 1 THEN ratings_1 + 1 WHEN OLD.rating = 1
    THEN ratings_1 - 1 ELSE ratings_1 END,
    ratings_2 = CASE WHEN NEW.rating = 2 THEN ratings_2 + 1 WHEN OLD.rating = 2 THEN
     ratings_2 - 1 ELSE ratings_2 END,
    ratings_3 = CASE WHEN NEW.rating = 3 THEN ratings_3 + 1 WHEN OLD.rating = 3 THEN
     ratings_3 - 1 ELSE ratings_3 END,
    ratings_4 = CASE WHEN NEW.rating = 4 THEN ratings_4 + 1 WHEN OLD.rating = 4 THEN
     ratings_4 - 1 ELSE ratings_4 END,
    ratings_5 = CASE WHEN NEW.rating = 5 THEN ratings_5 + 1 WHEN OLD.rating = 5 THEN
     ratings_5 - 1 ELSE ratings_5 END,
    average_rating = (average_rating*ratings_count - OLD.rating + NEW.rating)/
    ratings_count
    WHERE book_id = NEW.book_id;
    RETURN NEW;
END; $$ LANGUAGE 'plpgsql';

CREATE TRIGGER RatingUpdate
    AFTER UPDATE ON Ratings FOR EACH ROW XECUTE FUNCTION update();
```

## 3.3 Advanced Search

Our application provides the user with a special functionality to perform a custom search for books. The search is performed using the regular expression matching for book title and the authors if provided by the user and a regular conditional check on the year and rating intervals given by the user. Given, that there are more than one attributes to search on, we have defined a fixed format which the user should adhere to, so that we can extract the search attributes properly.

`Format:  Title|Author|Year(>,<)|Rating(>,<)`

```
def Advanced_Search_Query(title, author, year_1, year_2, rate_1, rate_2):
        query = "SELECT title FROM books WHERE true"
        if (len(title) != 0):
                query += " AND title LIKE '%{}%'".format(title)
        if (len(author) != 0):
                query += " AND authors LIKE '%{}%'".format(author)
        if (len(year_1) != 0):
                year_1 = int(year_1)
                query += " AND original_publication_year > {} ".format(year_1)
        if (len(year_2) != 0):
                year_2 = int(year_2)
                query += " AND original_publication_year < {}".format(year_2)
        if (len(rate_1) != 0):
                rate_1 = float(rate_1)
                query += "AND average_rating > {}".format(rate_1)
        if (len(rate_2) != 0):
                rate_2 = float(rate_2)
                query += "AND average_rating < {}".format(rate_2)
        return query
```

If we advance search for `"Potter|Rowling|1996,|,"` The query executed will be:

```
 SELECT title FROM books WHERE true
     AND title LIKE '%Potter%'
     AND authors LIKE '%Rowling%'
     AND original_publication_year >1996
```

## 3.4 Queries

```
--signin/signup page--
--1-- sign in verify id,password
SELECT * FROM users WHERE user_id=604 AND password='zdybeqnibm';
--2-- sign up
INSERT INTO users VALUES(53425,'abcdefghij');
--3-- Next User Id
SELECT MAX(user_id)+1 FROM users;

--user page--
--1-- Change Password button
UPDATE users SET password='abcdefghij' WHERE user_id=604;
--2-- Best Rated Books button
SELECT title FROM books ORDER BY average_rating DESC LIMIT 30;
--3-- Best Rated Authors button
SELECT author FROM authors ORDER BY rating DESC LIMIT 30;
--4-- Most Popular Books button
SELECT title FROM books ORDER BY ratings_count DESC LIMIT 30;
--5-- Most Popular Authors button
SELECT author FROM authors ORDER BY review_count DESC LIMIT 30;
--6-- Most Active Users button
SELECT user_id FROM ratings group by user_id ORDER BY count(rating) DESC LIMIT 30;
--7-- Most Recent Books button
SELECT title FROM books WHERE original_publication_year IS NOT null ORDER BY
    original_publication_year DESC, title LIMIT 30;
--8-- User Specific ToRead button
```

```sql
SELECT title FROM books, toread WHERE user_id=604 AND toread.book_id=books.book_id ORDER BY title;
--9-- User Rated Book button
SELECT title FROM books, ratings WHERE user_id=604 AND ratings.book_id=books.book_id ORDER BY rating DESC, title;
--10-- Search Book By Title
SELECT title FROM books WHERE title = 'The Alchemist';
--11-- Search Book By Author
SELECT title FROM books, (SELECT book_id, regexp_split_to_table(authors, ', ') AS ath FROM books) AS t1 WHERE books.book_id = t1.book_id AND t1.ath = 'J.K. Rowling';
--12-- Search Book By Tag
SELECT title FROM books, BookTags, tags WHERE tags.tag_name = 'favourites' AND tags.tag_id = BookTags.tag_id AND books.goodreads_book_id = booktags.goodreads_book_id ORDER BY average_rating DESC LIMIT 100;
--13-- Search Book By Year
SELECT title FROM books WHERE original_publication_year = 2017 ORDER BY average_rating DESC LIMIT 100;
--14-- Search Book By ISBN
SELECT title FROM booksW HERE isbn13 = 9780439023480;
--15-- Advanced Search
SELECT title FROM books WHERE true
        AND title LIKE '%Potter%'
        AND authors LIKE '%Rowling%'
        AND original_publication_year >1996
        AND average_rating > 2;

--book page--
--1-- Rate a Book
INSERT INTO ratings VALUES(604, 2, 3);
--2-- Change Rating
UPDATE ratings SET rating = 5 WHERE book_id = 2 AND user_id = 604;
--3-- Delete Rating
DELETE FROM ratings WHERE user_id=604 AND book_id=2;
--4-- Mark as toread
INSERT INTO ToRead VALUES(604, 88);
--5-- Remove toread
DELETE FROM ToRead WHERE user_id = 604 AND book_id = 88;
--6-- Get Book Info from books
SELECT title,authors,original_publication_year,isbn,language_code,average_rating,ratings_count,ratings_1,ratings_2,ratings_3,ratings_4,ratings_5,image_url,goodreads_book_id FROM books WHERE book_id = 6;
--7-- Get BookTag
SELECT tag_name FROM Tags, BookTags WHERE goodreads_book_id = 11870085 AND BookTags.tag_id = Tags.tag_id ORDER BY count DESC LIMIT 15;
--8-- Get user rating status
SELECT rating FROM ratings WHERE user_id=604 AND book_id=6;
--9-- Get toread status
SELECT * FROM ToRead WHERE user_id=604 AND book_id=6;

--author page--
--1-- Load author details from view
SELECT * FROM authors WHERE author = 'John Green';
--2-- Get all book info
SELECT title, image_url, count(user_id) FROM toread, books, (SELECT book_id, regexp_split_to_table(authors, ', ') AS ath FROM books) AS t1 WHERE toread.book_id = books.book_id AND books.book_id = t1.book_id AND t1.ath = 'John Green' GROUP BY books.book_id, toread.book_id ORDER BY average_rating DESC;

--admin page--
--1-- Next Book Id
SELECT MAX(book_id)+1 FROM books;
--2-- Add new Book
INSERT INTO books VALUES(10001,40522814,40522814,0,1,'125021615X',9781250216151,'K.M. Szpara',2020,'Docile','Docile','eng',3,1,1,1,0,0,1,0,0,'https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1557777604l/40522814.jpg','');
--3-- Get tag_id
SELECT tag_id FROM tags WHERE tag_name='fiction';
--4-- Add Tag
INSERT INTO booktags VALUES(40522814,11743,1);
--5-- Display List of Popular Tags
SELECT tag_name FROM tags, booktags WHERE tags.tag_id = booktags.tag_id GROUP BY tags.tag_id, booktags.tag_id ORDER BY SUM(booktags.count) DESC LIMIT 500;
```

## 3.5 Timing Report

The below table reports the execution timings of each of the queries averaged over three execution runs.

| **Query** | **Time To Execute** |
|---|---|
| Sign In/Sign Up Page | ———— |
| –1– sign in verify id/password | 96.378 ms |
| –2– sign up | 26.675 ms |
| –3– Next User Id | 30.040 ms |
| User Page | ———— |
| –1– Change Password | 25.212 ms |
| –2– Best Rated Books | 38.938 ms |
| –3– Best Rated Authors | 3.857 ms |
| –4– Most Popular Books | 9.482 ms |
| –5– Most Popular Authors | 2.634 ms |
| –6– Most Active Users | 7706.174 ms |
| –7– Most Recent Books | 6.225 ms |
| –8– User Specific ToRead | 156.011 ms |
| –9– User Rated Book | 66.244 ms |
| –10– Search Book By Title | 0.704 ms |
| –11– Search Book By Author | 35.426 ms |
| –12– Search Book By Tag | 333.394 ms |
| –13– Search Book By Year | 47.108 ms |
| –14– Search Book By ISBN | 30.895 ms |
| –15– Advanced Search | 26.477 ms |
| Book Page | ———— |
| –1– Rate a Book | 107.875 ms |
| –2– Change Rating | 10.989 ms |
| –3– Delete Rating | 11.134 ms |
| –4– Mark as toread | 11.114 ms |
| –5– Remove toread | 11.023 ms |
| –6– Get Book Info from books | 0.501 ms |
| –7– Get BookTag | 1.995 ms |
| –8– Get user rating status | 0.502 ms |
| –9– Get toread status | 0.590 ms |
| Author Page | ———— |
| –1– Load author details from view | 1.639 ms |
| –2– Get all book info | 125.662 ms |
| Moderator Page | ———— |
| –1– Next Book Id | 0.294 ms |
| –2– Add new Book | 70.555 ms |
| –3– Get tag_id | 20.489 ms |
| –4– Add Tag | 35.169 ms |
| –5– Display List of Popular Tags | 426.825 ms |
| Create Index Query [section 3.2.2] | 597.491 ms |
| Create Materialized View [section 3.2.3] | 754.896 ms |
| Create Insert Function | 61.050 ms |
| Trigger Rating Insert | 11.067 ms |
| Create Delete Function | 36.622 ms |
| Trigger Rating Delete | 11.081 ms |
| Create Update Function | 44.504 ms |
| Trigger Rating Update | 11.074 ms |

Table 5: Query execution times

## 3.6 Front-End View

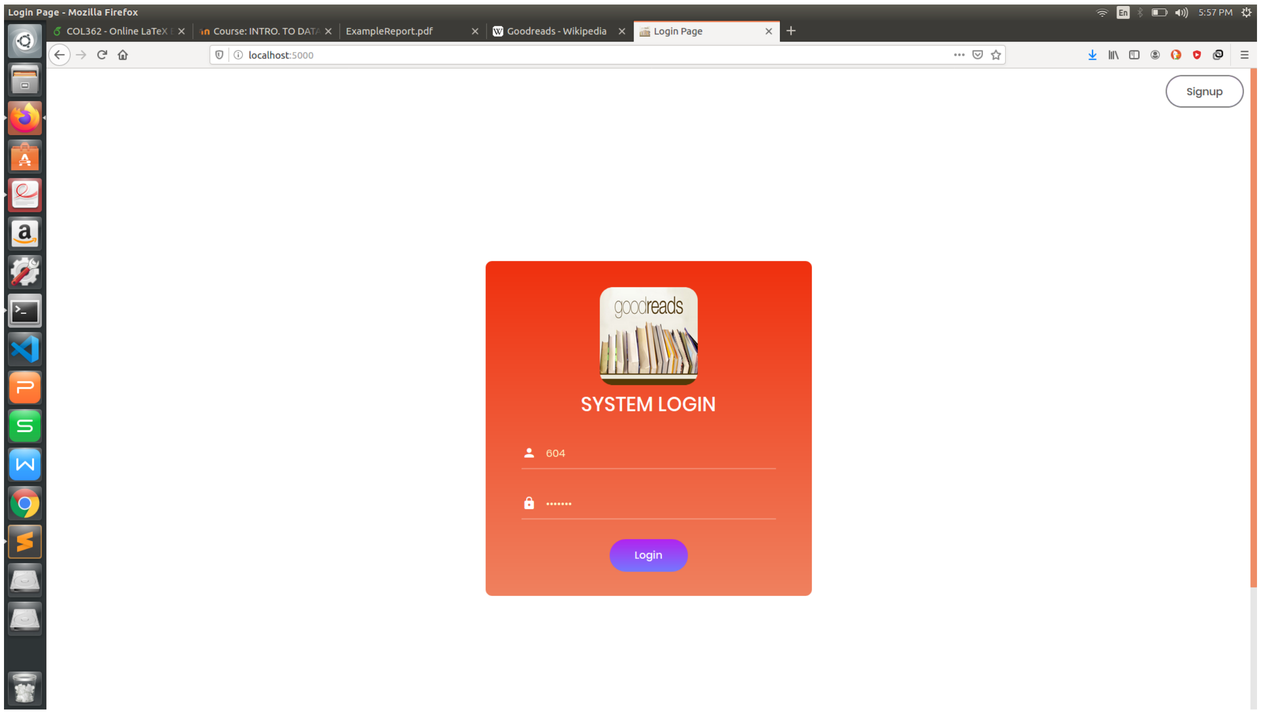
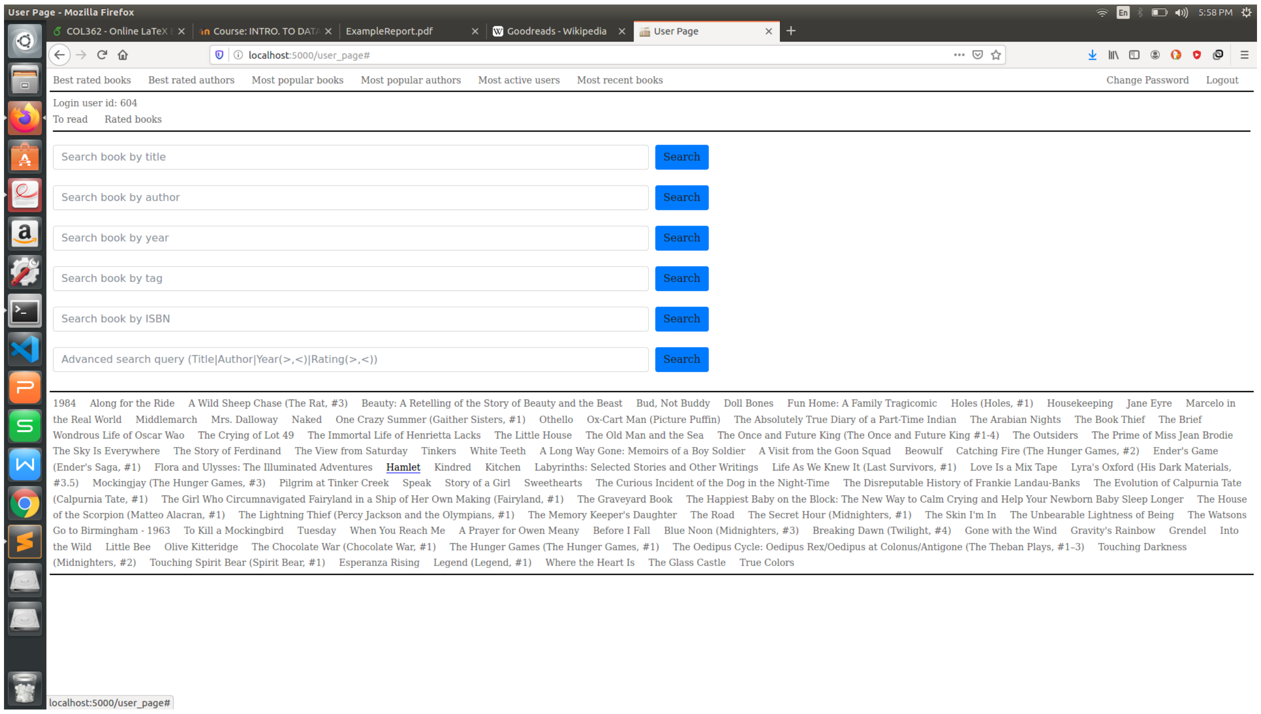


Figure 2: Sign up/Login Page

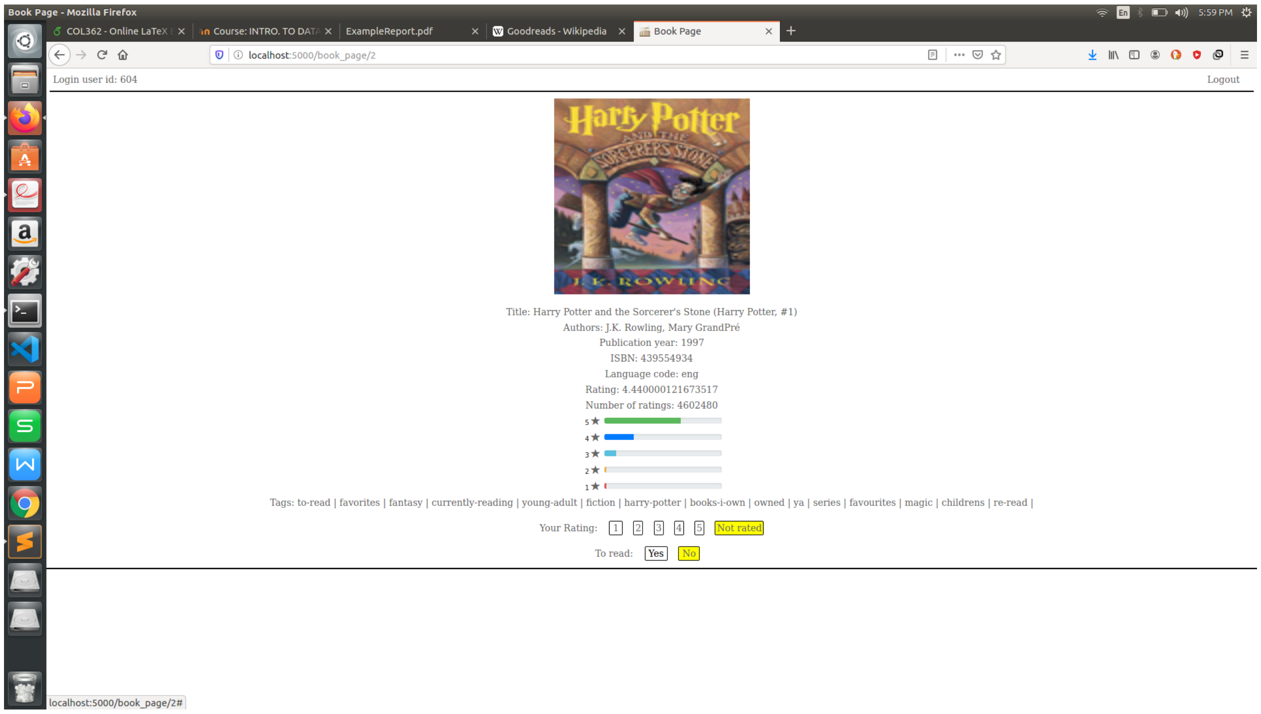


Login user id: 604

To read    Rated books

1984    Along for the Ride    A Wild Sheep Chase (The Rat, #3)    Beauty: A Retelling of the Story of Beauty and the Beast    Bud, Not Buddy    Doll Bones    Fun Home: A Family Tragicomic    Holes (Holes, #1)    Housekeeping    Jane Eyre    Marcelo in the Real World    Middlemarch    Mrs. Dalloway    Naked    One Crazy Summer (Gaither Sisters, #1)    Othello    Ox-Cart Man (Picture Puffin)    The Absolutely True Diary of a Part-Time Indian    The Arabian Nights    The Book Thief    The Brief Wondrous Life of Oscar Wao    The Crying of Lot 49    The Immortal Life of Henrietta Lacks    The Little House    The Old Man and the Sea    The Once and Future King (The Once and Future King #1-4)    The Outsiders    The Prime of Miss Jean Brodie    The Sky Is Everywhere    The Story of Ferdinand    The View from Saturday    Tinkers    White Teeth    A Long Way Gone: Memoirs of a Boy Soldier    A Visit from the Goon Squad    Beowulf    Catching Fire (The Hunger Games, #2)    Ender's Game (Ender's Saga, #1)    Flora and Ulysses: The Illuminated Adventures    Hamlet    Kindred    Kitchen    Labyrinths: Selected Stories and Other Writings    Life As We Knew It (Last Survivors, #1)    Love Is a Mix Tape    Lyra's Oxford (His Dark Materials, #3.5)    Mockingjay (The Hunger Games, #3)    Pilgrim at Tinker Creek    Speak    Story of a Girl    Sweethearts    The Curious Incident of the Dog in the Night-Time    The Disreputable History of Frankie Landau-Banks    The Evolution of Calpurnia Tate (Calpurnia Tate, #1)    The Girl Who Circumnavigated Fairyland in a Ship of Her Own Making (Fairyland, #1)    The Graveyard Book    The Happiest Baby on the Block: The New Way to Calm Crying and Help Your Newborn Baby Sleep Longer    The House of the Scorpion (Matteo Alacran, #1)    The Lightning Thief (Percy Jackson and the Olympians, #1)    The Memory Keeper's Daughter    The Road    The Secret Hour (Midnighters, #1)    The Skin I'm In    The Unbearable Lightness of Being    The Watsons Go to Birmingham - 1963    To Kill a Mockingbird    Tuesday    When You Reach Me    A Prayer for Owen Meany    Before I Fall    Blue Noon (Midnighters, #3)    Breaking Dawn (Twilight, #4)    Gone with the Wind    Gravity's Rainbow    Grendel    Into the Wild    Little Bee    Olive Kitteridge    The Chocolate War (Chocolate War, #1)    The Hunger Games (The Hunger Games, #1)    The Oedipus Cycle: Oedipus Rex/Oedipus at Colonus/Antigone (The Theban Plays, #1–3)    Touching Darkness (Midnighters, #2)    Touching Spirit Bear (Spirit Bear, #1)    Esperanza Rising    Legend (Legend, #1)    Where the Heart Is    The Glass Castle    True Colors

Figure 3: User Page



Login user id: 604

Title: Harry Potter and the Sorcerer's Stone (Harry Potter, #1)

Authors: J.K. Rowling, Mary GrandPré

Publication year: 1997

ISBN: 439554934

Language code: eng

Rating: 4.440000121673517

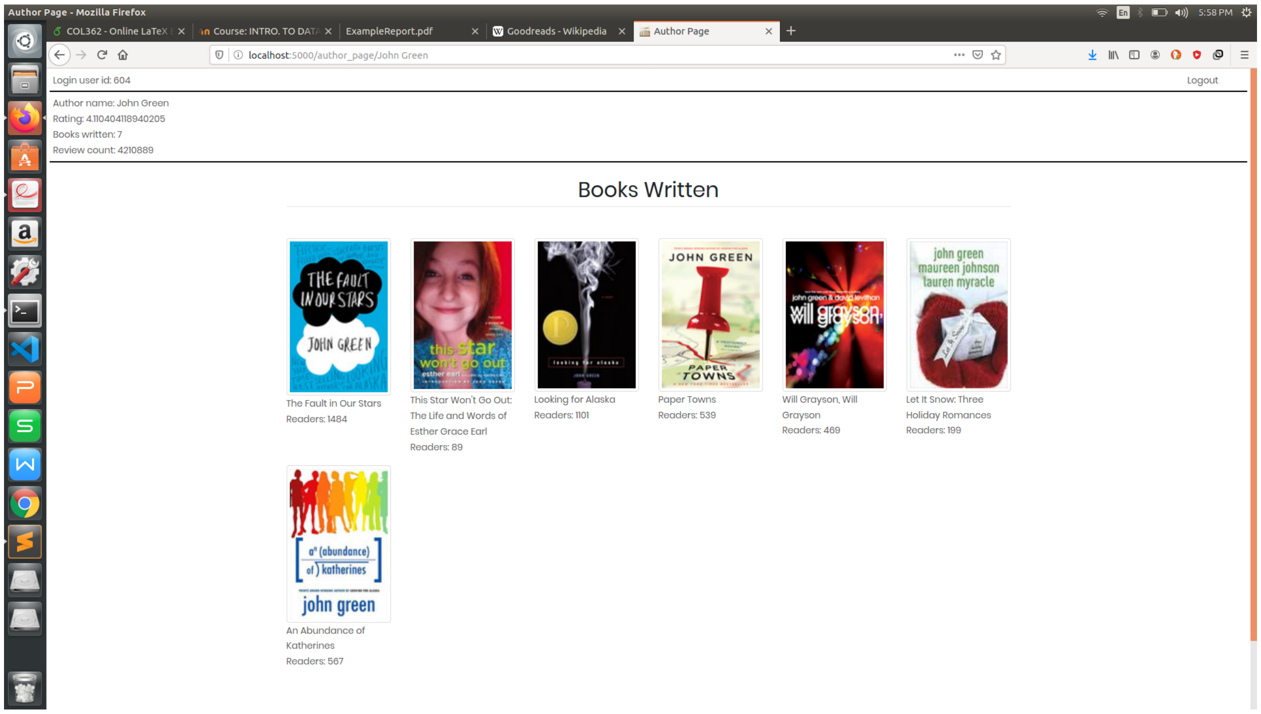
Number of ratings: 4602480

Tags: to-read | favorites | fantasy | currently-reading | young-adult | fiction | harry-potter | books-i-own | owned | ya | series | favourites | magic | childrens | re-read |

Your Rating: 1 2 3 4 5 [✓ Not rated]

To read: Yes [✓ No]

Figure 4: Book Page



Author Page - Mozilla Firefox

COL362 - Online LaTeX | Course: INTRO. TO DATA | ExamplePDF.pdf | Goodreads - Wikipedia | Author Page

localhost:5000/author_page/John Green

Login user id: 604 Logout

Author name: John Green
Rating: 4.110404118940205
Books written: 7
Review count: 4210889

Books Written

The Fault in Our Stars
Readers: 1484

This Star Won't Go Out: The Life and Words of Esther Grace Earl
Readers: 89

Looking for Alaska
Readers: 1101

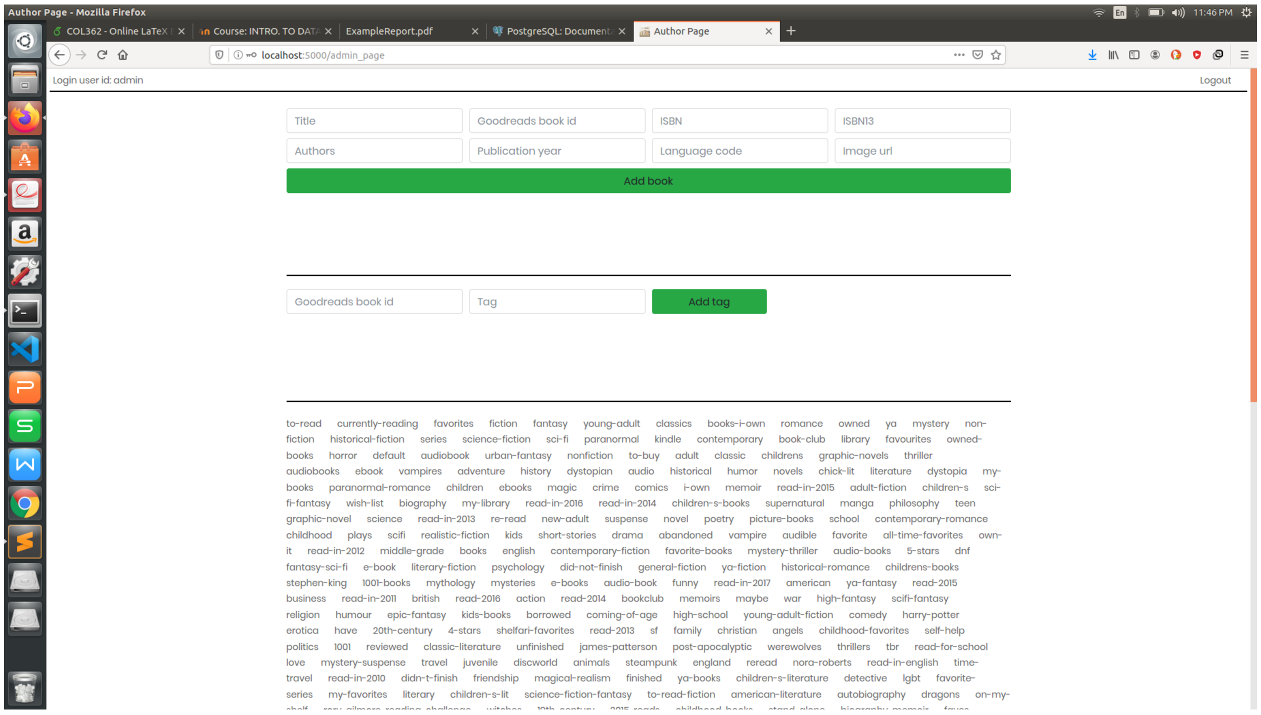
Paper Towns
Readers: 539

Will Grayson, Will Grayson
Readers: 469

Let It Snow: Three Holiday Romances
Readers: 199

An Abundance of Katherines
Readers: 567

Figure 5: Author Page



Author Page - Mozilla Firefox

COL362 - Online LaTeX | Course: INTRO. TO DATA | ExampleReport.pdf | PostgreSQL: Document | Author Page

localhost:5000/admin_page

Login user id: admin Logout

Title | Goodreads book id | ISBN | ISBN13
Authors | Publication year | Language code | Image url

Add book

Goodreads book id | Tag | Add tag

to-read currently-reading favorites fiction fantasy young-adult classics books-i-own romance owned ya mystery non-fiction historical-fiction series science-fiction sci-fi paranormal kindle contemporary book-club library favourites owned-books horror default audiobook urban-fantasy nonfiction to-buy adult classic childrens graphic-novels thriller audiobooks ebook vampires adventure history dystopian audio historical humor novels chick-lit literature dystopia my-books paranormal-romance children ebooks magic crime comics i-own memoir read-in-2015 adult-fiction children-s sci-fi-fantasy wish-list biography my-library read-in-2016 read-in-2014 children-s-books supernatural manga philosophy teen graphic-novel science read-in-2013 re-read new-adult suspense novel poetry picture-books school contemporary-romance childhood plays scifi realistic-fiction kids short-stories drama abandoned vampire audible favorite all-time-favorites own-it read-in-2012 middle-grade books english contemporary-fiction favorite-books mystery-thriller audio-books 5-stars dnf fantasy-sci-fi e-book literary-fiction psychology did-not-finish general-fiction ya-fiction historical-romance childrens-books stephen-king 1001-books mythology mysteries e-books audio-book funny read-in-2017 american ya-fantasy read-2015 business read-in-2011 british read-2016 action read-2014 bookclub memoirs maybe war high-fantasy scifi-fantasy religion humour epic-fantasy kids-books borrowed coming-of-age high-school young-adult-fiction comedy harry-potter erotica have 20th-century 4-stars shelfari-favorites read-2013 sf family christian angels childhood-favorites self-help politics 1001 reviewed classic-literature unfinished james-patterson post-apocalyptic werewolves thrillers tbr read-for-school love mystery-suspense travel juvenile discworld animals steampunk england reread nora-roberts read-in-english time-travel read-in-2010 didn-t-finish friendship magical-realism finished ya-books children-s-literature detective lgbt favorite-series my-favorites literary children-s-lit science-fiction-fantasy to-read-fiction american-literature autobiography dragons on-my-

Figure 6: Moderator Page