

Fenyx IT Academy

Database Project

November 2022

Library CLI Application



Overview

This project aims to create a library system. Let's imagine a library:

- You can give books to this library.
- You can register to this library and keep track of the books you read/liked.
- You can find the books by their genre/author.
- You can see the top books/authors.

With this project, you will be able to have all these functionalities in your Library Command-line application.

Sample Scenarios

Scenario 1

To illustrate, a person who wants to borrow the “Harry Potter” book from the library needs to first register at the library. Thus he will enter the command **sign_up**, only username is required for sign up. After signing up, he will enter the command **borrow_book** and will give the “Book ID” together with this command. After running the **borrow_book** command if this book is available in the library he will be able to borrow the book. Later he can run the command **return_book** to return this book. A user can also mark books as read by running **mark_read** command or add this book to his/her favorites by running **fav_book** command. Both commands will require “Username” and “Book ID”.

Scenario 2

To illustrate, everyone can add books to the library by running the **add_book** command. It is possible to search books by their name or their genre which is described in more detail in the [Command Details](#) section. By executing these commands, you can also see detailed information about the books including their “Book ID”.

Typer

You will use the [Typer](#) library in the implementation of this project. Typer is a library for building CLI applications that users will love using and developers will love creating. You can follow [this tutorial](#) to get familiar with the library.

Example repository we created for you → <https://github.com/iremugurlu/sample-library-cli-app>

Command Details

Following commands are available to all users.

1. **start**

This command doesn't take any arguments. It creates the database if not created. If it is created, it connects to it. Finally, it logs the following output.

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py start
Welcome to Library CLI!
```

```
You can execute command '--help' to see the possible commands
```

2. *sign_up*

This command takes ‘NAME’ as an argument. It adds a user to the database with his username. If the username is already in use, logs an error message and asks for another username.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py sign_up John
Nice that you are signing up!
This username already exists!
Please enter another username: Michael
Successfully signed up!
```

3. *add_book*

This command doesn’t take any arguments. It asks for the details of the book: *name, author, # pages* and *genre*. Then it adds the book to the database. The books are identical by the combination of their name and author. If another book with the same name and author exists, increment the quantity of the existing book instead of adding a new one.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py add_book
Please enter the required book info to add!
Name: The Hunger Games
Author: Suzanne Collins
# Pages: 374
Genre: Science fiction
Successfully added book!
```

4. *search_by_name*

This command takes ‘NAME’ as an argument. It displays books with this name in the table view.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py search_by_name 'Animal Farm'
| # | Book ID | Name | Author | # Pages | Genre | Availability |
|---|---|---|---|---|---|---|
| 1 | 1 | Animal Farm | George Orwell | 140 | Animal fable | True |
```

5. *search_by_author*

This command takes ‘AUTHOR’ as an argument. It displays books written by this author in the table view.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py search_by_author 'Stephen King'
| # | Book ID | Name | Author | # Pages | Genre | Availability |
|---|---|---|---|---|---|---|
| 1 | 4 | The Outsider | Stephen King | 496 | Thriller | True |
| 2 | 7 | Fairy Tale | Stephen King | 607 | Thriller | True |
```

6. *recently_added*

This command takes ‘GENRE’ as an **optional** argument. If it is provided, it displays the 5 most recent books added in a given genre. If not, it displays the 5 most recent books added. **Note:** Keep the “date_added” information when adding a book so you can identify the most recent ones.

Without genre

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py recently_added
```

#	Book ID	Name	Author	# Pages	Genre	Availability
1	9	The Hunger Games	Suzanne Collins	374	Science fiction	True
2	8	1984	George Orwell	328	Science fiction	True
3	7	Fairy Tale	Stephen King	607	Thriller	True
4	6	The Girl on the Train	Paula Hawkins	336	Thriller	True
5	5	The Great Gatsby	F. Scott Fitzgerald	110	Romance	True

With genre

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py recently_added Thriller
```

#	Book ID	Name	Author	# Pages	Genre	Availability
1	7	Fairy Tale	Stephen King	607	Thriller	True
2	6	The Girl on the Train	Paula Hawkins	336	Thriller	True
3	4	The Outsider	Stephen King	496	Thriller	True

7. *most_read_books*

This command takes ‘GENRE’ as an **optional** argument. If it is provided, it displays the 10 most read books in a given genre. If not, it displays the 10 most read books.

Without genre

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py most_read_books
```

#	Book ID	Name	Author	Genre	Count
1	7	Fairy Tale	Stephen King	Thriller	3
2	1	Animal Farm	George Orwell	Animal fable	3
3	4	The Outsider	Stephen King	Thriller	2
4	6	The Girl on the Train	Paula Hawkins	Thriller	1
5	8	1984	George Orwell	Science fiction	1
6	9	The Hunger Games	Suzanne Collins	Science fiction	1
7	2	Invisible Man	Ralph Ellison	Novel	1
8	5	The Great Gatsby	F. Scott Fitzgerald	Romance	1

With genre

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py most_read_books Thriller
```

#	Book ID	Name	Author	Genre	Count
1	7	Fairy Tale	Stephen King	Thriller	3
2	4	The Outsider	Stephen King	Thriller	2
3	6	The Girl on the Train	Paula Hawkins	Thriller	1

8. *most_favorite_books*

This command takes ‘GENRE’ as an **optional** argument. If it is provided, it displays the 10 most favorite books in a given genre. If not, it displays the 10 most favorite books.

Without genre

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py most_favorite_books
```

#	Book ID	Name	Author	Genre	Count
1	3	The Kite Runner	Khaled Hosseini	Novel	2
2	5	The Great Gatsby	F. Scott Fitzgerald	Romance	2
3	2	Invisible Man	Ralph Ellison	Novel	1
4	1	Animal Farm	George Orwell	Animal fable	1

With genre

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py most_favorite_books Novel
```

#	Book ID	Name	Author	Genre	Count
1	3	The Kite Runner	Khaled Hosseini	Novel	2
2	2	Invisible Man	Ralph Ellison	Novel	1

9. *most_read_genres*

This command doesn’t take any arguments. It displays the 5 most read genres.

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py most_read_genres
```

#	Genre	Count
1	Thriller	6
2	Animal fable	3
3	Science fiction	2
4	Novel	1
5	Romance	1

10. *most_read_authors*

This command doesn't take any arguments. It displays the 3 most read authors.

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py most_read_authors
```

#	Author	Count
1	Stephen King	5
2	George Orwell	4
3	F. Scott Fitzgerald	1

Following commands require username and user sign up:

11. *borrow_book*

This command takes 'BOOK ID' and 'USERNAME' as arguments. If this book is available, saves the data as this user borrowed the book and reduces the available amount of the book. If not available, then logs an error message saying this book is not available.

Available book

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py borrow_book 3 Ceren  
You borrowed book 3!
```

Unavailable book

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py borrow_book 3 Ceren  
Sorry, book 3 is not available! Try again later.
```

12. *return_book*

This command takes 'BOOK ID' and 'USERNAME' as arguments. If this user borrowed the book previously, saves the data as this user returned the book and increments the available amount of the book. If this user did not borrow the book, then logs an error message saying this book is not borrowed by him.

Borrowed book

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py return_book 3 Ceren  
You returned book 3!
```

Not borrowed book

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py return_book 4 Ceren  
Sorry, you didn't borrow book 4.
```

13. *mark_read*

This command takes ‘BOOK ID’ and ‘USERNAME’ as arguments. It marks this book as “read” for this user.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py mark_read 5 Ceren  
You marked book 5 as read!
```

14. *mark_reading*

This command takes ‘BOOK ID’ and ‘USERNAME’ as arguments. It marks this book as “reading” for this user.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py mark_reading 1 Ceren  
You marked book 1 as reading!
```

15. *mark_will_read*

This command takes ‘BOOK ID’ and ‘USERNAME’ as arguments. It marks this book as “will read” for this user.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py mark_will_read 3 Ceren  
You marked book 3 as will read!
```

16. *fav_book*

This command takes ‘BOOK ID’ and ‘USERNAME’ as arguments. It adds this book to this user’s favorites.

```
|ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py fav_book 5 Irem  
You added book 5 to your favorites!
```

17. *my_books*

This command takes ‘USERNAME’ as an argument. It displays the books you read, are reading, will read and your favorite books.

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py my_books Ceren
BOOKS YOU READ
```

#	Book ID	Name	Author	# Pages	Genre	Availability
1	5	The Great Gatsby	F. Scott Fitzgerald	110	Romance	True

BOOKS YOU ARE READING

#	Book ID	Name	Author	# Pages	Genre	Availability
1	1	Animal Farm	George Orwell	140	Animal fable	True

BOOKS YOU WILL READ

#	Book ID	Name	Author	# Pages	Genre	Availability
1	3	The Kite Runner	Khaled Hosseini	348	Novel	True

YOUR FAVORITE BOOKS

#	Book ID	Name	Author	# Pages	Genre	Availability

18. *statistics*

This command takes ‘USERNAME’ as an argument. It displays the following statistics in a table: *number of books you read, number of authors you read, number of genres you read, number of total pages you read.*

```
ceren.ugurlu@ceren-ugurlu-C02YJ615JGH6 book-store-cli-app % python3 main.py statistics Ceren
```

Statistic	Number
Books you read	6
Authors you read	5
Genres you read	4
Total pages you read	2496

Definition of Done

- Full attendance at mentor meetings
- ERD Diagram
- Python CLI application
- Project presentation
- GitHub repository with a README

General Requirements

- **PostgreSQL** will be used as a database management system.
- **GitHub** will be used in the project.
 - Each team will have a GitHub repository and each team member will be added as a collaborator.
- **Trello Board** will be used in the project.
 - Team mentors will be added to the board and they will check if the team uses Trello actively.
- **Meetings**
 - There will be at least 30-mins meeting with teammates every day. The content of the daily meeting is generally as follows: what each teammate has done, the general direction of the project and task sharing until tomorrow.
 - Each team will have a mentor. A meeting will be held with team mentor and team members on the specified dates (once in 2 - 3 days). The time of the meetings can be determined by each team.
- **Presentation** at the end of the project. To complete the project, all members have to show up in the presentation and present the program.

Instruction Steps

You have to stick to the schedule. You will have a progress meeting with your mentor once in 2-3 days. You have to complete related steps before the next meeting.

Step 1

- Project Kick-off Meeting
- Reading instructions
- Understanding and discussing the requirements of the project with your teammates

Step 2

Database & ERD Design

- Design the database according to the requirements of the project.
- You can use the [pgAdmin ERD tool](#) to draw your ERD diagram.

Note: Present your DB design and ERD diagram to your mentor so you can get feedback about it!

Step 3

- Extract the SQL query from your ERD design to create the necessary tables.
- Use this SQL file in your python application to create tables in your database upon the application start.

Note: [Step 4](#) and [Step 5](#) can be done in parallel by different teammates!

Step 4

Implement all commands described in the [Command Details](#) section.

Step 5

Write SQL queries to get/add the required information from/to the database.

Tip: Having a separate **database.py** file to connect to database and execute queries would ease your work!

Step 6

Connect *commands* and *database functions* to have a running app.

Step 7

Test Your Program

- Test your program and try to find the bugs.
- Execute all of the commands your program has and verify that they work correctly.
- Try different scenarios to discover new bugs.
- Improve the exception handling in your program. Show useful error messages to users when undesired behavior occurs.