Biblio

Logo

Documentation

issues repo not found forks repo not found stars repo not found license repo not found

DB Postgres Container Docker

WebFramework Flask Framework Bootstrap

Language Python Language HTML Language CSS

Biblio is a tool for your private library. It manages your books and magazines.

Composed Docker Containers

Otherwise, you can use the docker-image provided in the repository.

The first time you use this docker-compose you must initialise the containers with:

```
cd biblio
docker-compose up
```

The Frontend is now visible under `localhost:5000`

Afterwards, the container may be started with

```
docker-compose start
```

To shut down the container you can either use a different terminal and use

```
docker-compose stop
```

or by pressing Ctrl + C in the terminal used to initialise the container.

About the project

Team

- Lukas Benner (6550912)
- Phillip Lange (5920414)
- Alina Buss ()

Target

With Biblio we want to build a system to keep track of your private book collection like its a real library.

Biblio is your tool to manage your own private book collection. Whether you want to keep track of all your books. Find gems you forgot about or keep track of your reading list. With Biblio you can also invite friends and family to your private library. It allows you to share and borrow books between each other.

We think its important to know your possessions and keep it managed. We can help you to focus on reading rather than searching. On the other hand its very important to share, so everybody can enjoy reading and learning new stuff. Biblio helps you to organize this process easily.

Tools

For developing Biblio we used Python, Flask, Postgres and a little bit of bootstrap for easier styling.

Presentation

Screenshot
Screenshot
Screenshot

Specification

'Biblio' will be a webapp for organizing your private book collection.

Group-member: Alina Buss (4163246), Phillip Lange (5920414), Lukas Benner (6550912)

The used database schema und functionalities are described in the following:

- Books: Every book is stored in the database with following attributes: title, ISBN, recommended Age, language, publishing date, genre, edition, wether its available and a unique Identifier.

- Author & Publisher: Every Book has a authoter and Publishier assigned. These contain information like address, name and a unique identifier.

- Location: Every book has a location assigned, where its supossed to be. The Location has the following attributes: address, floor, room, shelf, compartment and a unique identifier.

- User: Every user of the database has a name, a user_id and his/her date of birth is stored.

- Borrow: A User can lend severall books. For every loan the timestamp will be stored. For every borrowed item the duration and the Book_ID will be used to assign it to a loan.

- The database can tracks which books a particular user already read.

- The Address is handled in a seperate table and has the following attributes: city, country, street, house number, zipcode and a unique identifier.

ER-Model

Screenshot

Relational Model

Screenshot

Database Modeling Explanation

1. atomar attributes

- There are only Single Valued Attributes.
- Attribute Domain does not change.
- There is a Unique name for every Attribute/Column.
- The order in which data is stored, does not matter.

  Therefore we did following:

  - split address
  - split names
  - allowed several authors
  - unique names inside a table

2. remove the repeated information

- Second Normal Form (2NF) is based on the concept of full functional dependency.

  Therefore we did following:

  - delete address attributes
  - make separate address table

3. No non-primary-key attribute is transitively dependent on the primary key

- A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

  Therefore, we did following:

  - made separate tables for author, publisher and connected it with the books table

Application

Video

The video can be found on YouTube under: https://youtu.be/rB1QyN_s4V4

Links

- Queries (more can be found in the code)
- SQL-File
- Github