CMPE321 – Summer/2020

Library Database

Sertay Akpınar

Introduction

In this project, I implemented a small library database with a simple web-based user interface using flask in python. I made some assumptions and design choices in order to make the system efficient. Additionally, I clarified the tables and the DML operations I used. Last but not least, I added the input/output screenshots of the system.

Relation Definitions

Books

It stores the isbn, title and author of a book. It's like a simple library.

Field	Type	Null	Key	Default	Extra
isbn	int	NO	PRI		auto_increment
title	varchar(100)	YES			
author	varchar(50)	YES			

Isbn → title, author

Borrowed

It stores the isbn of a book, tc of the user who borrowed the book and the due date. It's for keeping track of the people who borrowed the books, and the books which are borrowed.

Field	Type	Null	Key	Default	Extra
isbn	int	YES			
tc	bigint	YES			
due_date	timestamp	YES			

Users

It stores the user id, tc and total number of books of each user. This table is useful for keeping track of the number of borrowed books of each user.

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI		auto_increment
tc	bigint	YES			
total_books	int	YES		1	

Id → tc, total_books

It's in conformity with BCNF.

SQL DML

```python
@app.route('/add_book', methods=['GET', 'POST'])
def add_book():
    form = BookForm(request.form)
    if request.method == 'POST' and form.validate():
        title = form.title.data
        author = form.author.data

        # Create Cursor
        cur = mysql.connection.cursor()

        # Execute
        cur.execute("INSERT INTO books(title, author) VALUES(%s, %s)", (title, author))

        # Commit to DB
        mysql.connection.commit()

        # Close connection
        cur.close()

        flash('Book Created', 'success')
        return redirect(url_for('books'))

    return render_template('add_book.html', form=form)
```

I used INSERT operation to add a book to the table.

```python
    if {'isbn': isbn} not in isbn_borrowed:
        if {'isbn': isbn} in isbn_books:
            if total_book_num < 8:
                cur.execute("INSERT INTO borrowed(isbn, tc, due_date) VALUES(%s, %s, %s)", (isbn, tc, due_date))
                if {'tc': tc} not in tc_numbers:
                    cur.execute("INSERT INTO users(tc) VALUES({0})".format(tc))
                else:
                    cur.execute("UPDATE users SET total_books = total_books + 1 WHERE tc={0}".format(tc))
                mysql.connection.commit()
                flash('Book is borrowed', 'success')
            else:
                flash('User reached the total borrowed books limit!', 'danger')
        else:
            flash("Book's isbn not found!", 'danger')
    else:
        flash('Book is already borrowed by another user!', 'danger')

    # Close connection
    cur.close()
    return redirect(url_for('books'))

return render_template('borrow_book.html', form=form)
```

I used UPDATE operation to increment the value of total books of a user, after borrowing the book.

```python
@app.route('/delete_book/<string:isbn>', methods=['POST'])
def delete_book(isbn):
    # Create cursor
    cur = mysql.connection.cursor()

    # Execute
    cur.execute("DELETE FROM books WHERE isbn = %s", [isbn])

    # Commit to DB
    mysql.connection.commit()

    # Close connection
    cur.close()

    flash('Book Deleted', 'success')

    return redirect(url_for('books'))
```

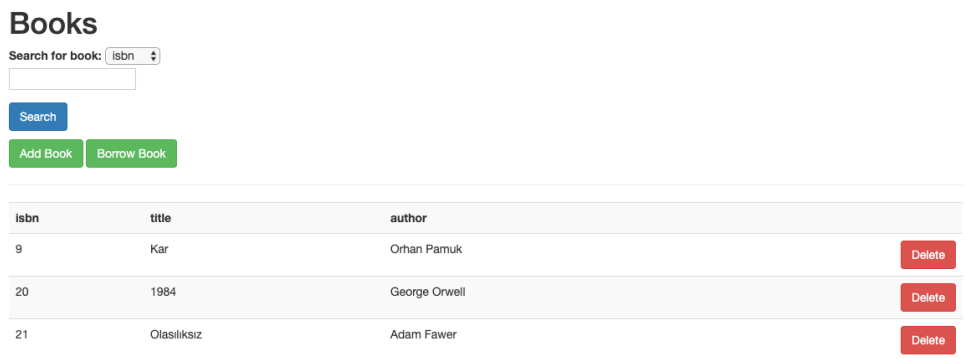
I used DELETE operation to for deleting the book.

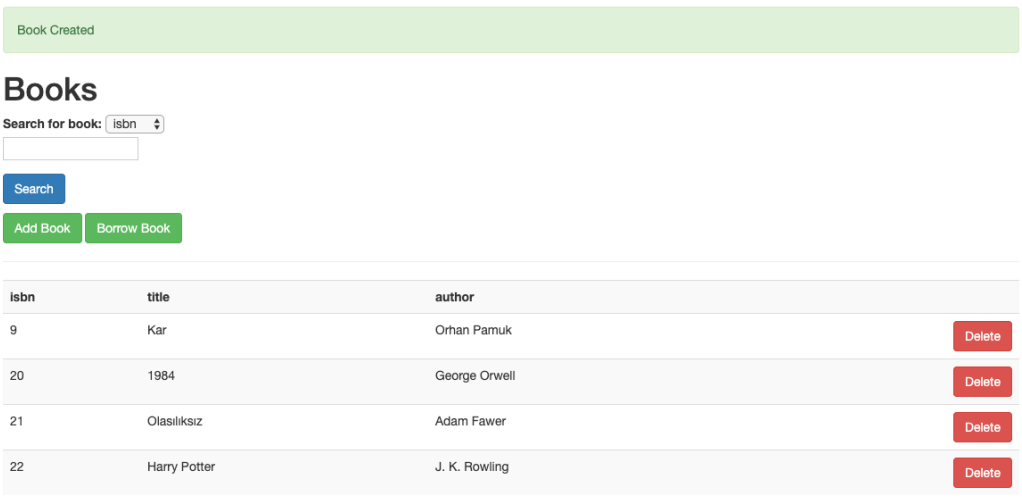
Constraints

- The only borrowable items are books in the library.
- Borrowers should be able to search books according to title, ISBN, author.
- Users can borrow books with TC number (11 decimal).
- Each person may borrow book for two weeks.
- Each person may borrow at most 8 books.
- The number of books a person has borrowed, and their due dates should be accessible.

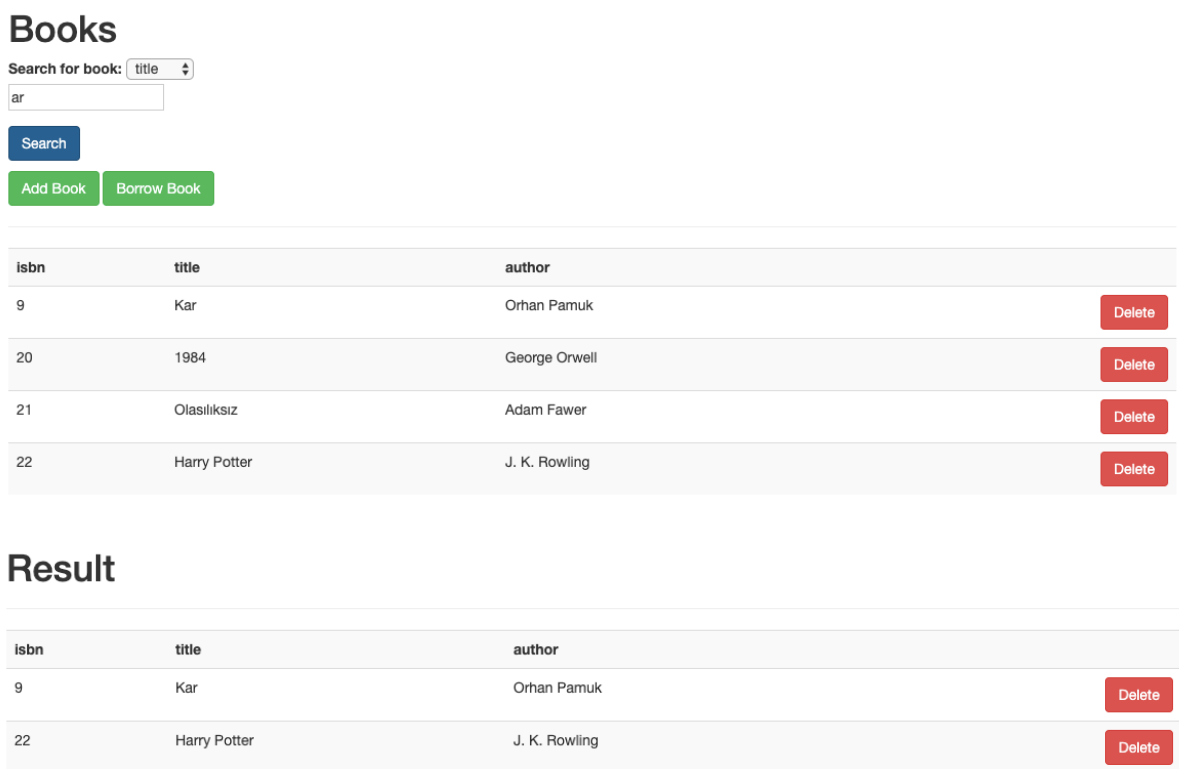
Input and Output

Create Book:

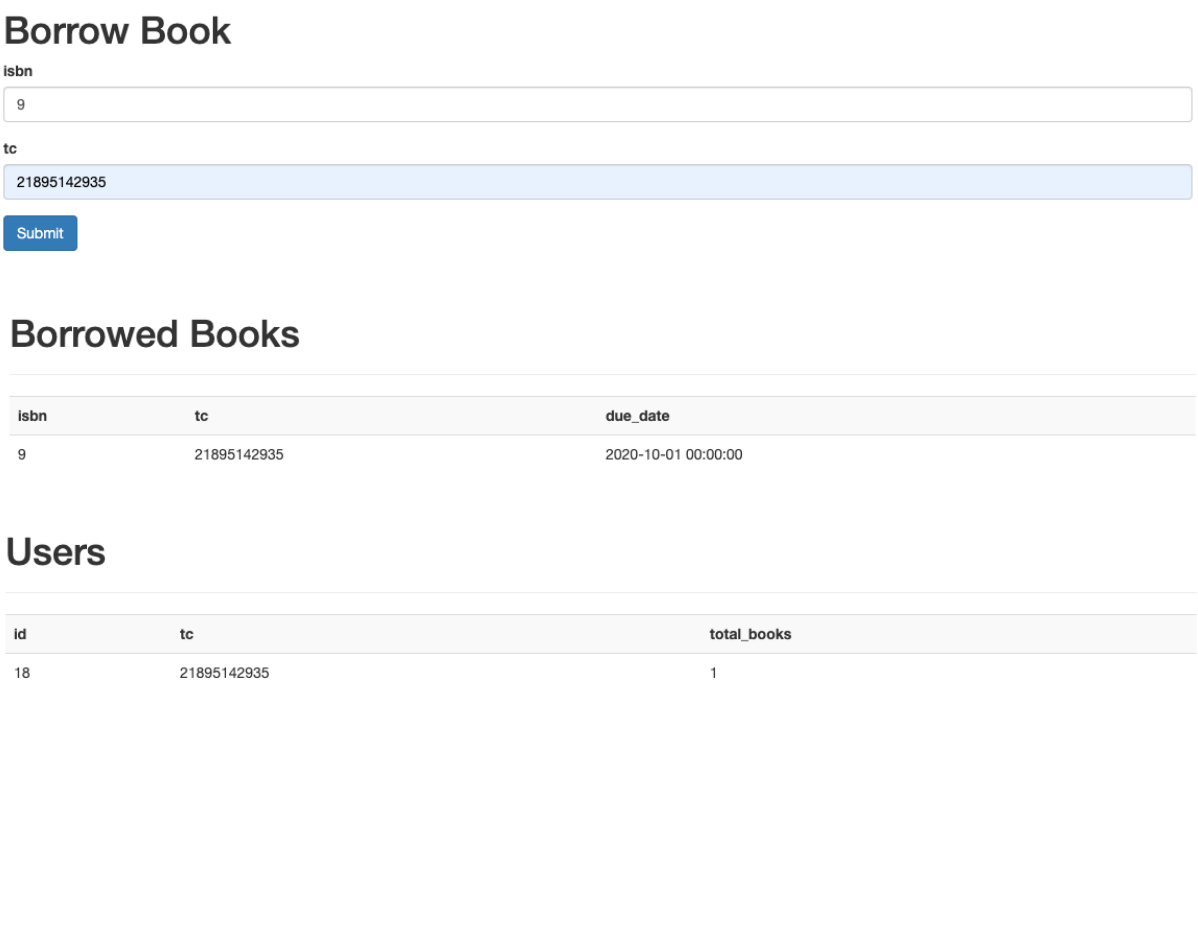




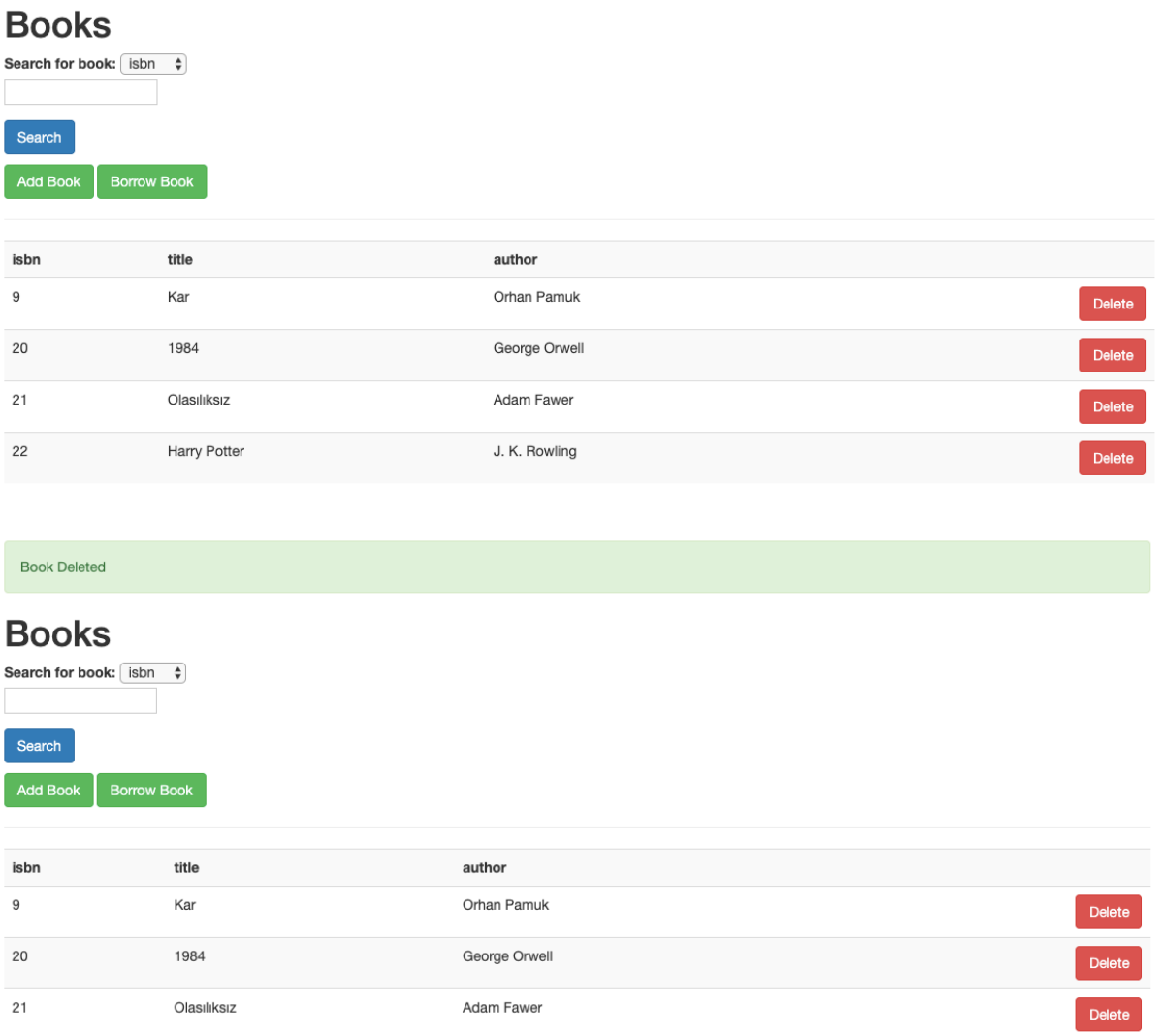
Search Book:



Borrow Book:



Delete Book:



Conclusion:

In this project, I have implemented a small library database with a simple web-based user interface using flask in python. I also documented my app according to the constraints mentioned in the project description. My main goal while doing this project was to understand the logic behind the database management system and have knowledge on the structures and the algorithms I used. At the same time, I have tried to design my system as efficiently as possible. However, I believe that there some certain things need to be improved for future development. One of them is, creating a new login/register layer to the system to improve the application. In this way, each person has an account and each of them can access the system on its own.