HelloWorld High Library Management System:

User Guide

URL: http://127.0.0.1:8000/

**NOTE: This documentation is only useful after the library application itself has been installed, install instructions are located here*

Table of Contents:

Login Page

Install the Library Application here, (if it is already installed skip this) then go to this URL to access the application:

http://127.0.0.1:8000/

You will be greeted with a login page. There are 2 types of users that can log in:

Students:

- ✓ Can view Books in the HelloWorld High Library

Library Administrators & Faculty/Teachers:

- ✓ Can Edit/View/Add/Delete books
- ✓ Can Edit/View/Add/Delete users
- ✓ Can see recent actions (history)

Library Administrator logins that may be used for testing purposes are:

USERNAME:	PASSWORD:
gat001admin	xxx12345
job001admin	xxx12345
sna001admin	xxx12345

Additionally, there is a user called the Superuser that may edit and maintain the site itself and has unlimited access

USERNAME:	PASSWORD:
admin	xxx12345

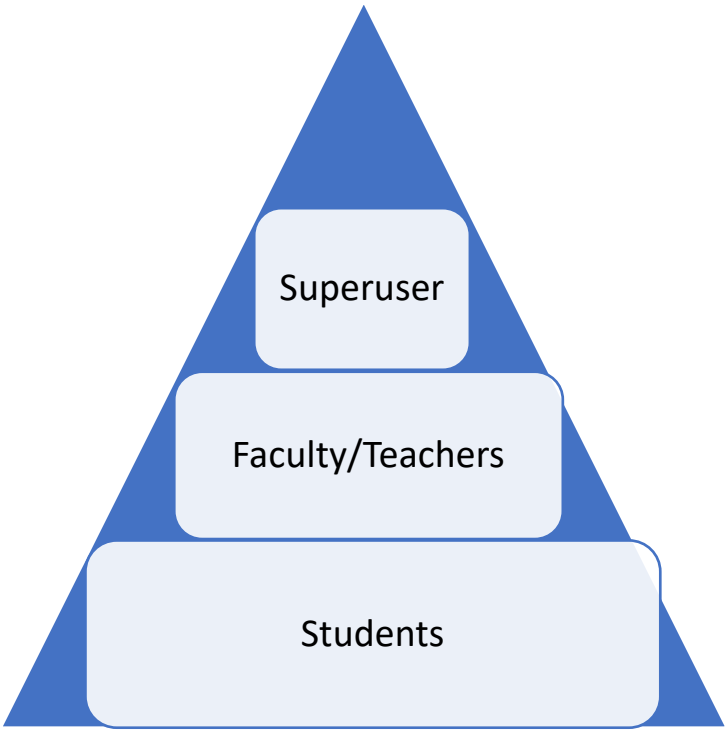
Student logins that may be used for testing purposes are:

Note they are strictly limited to viewing, no changes may be made as a student

ant001	xxx12345
bla001	xxx12345
fer001	xxx12345
gar001	xxx12345
hag001	xxx12345

pri001	xxx12345

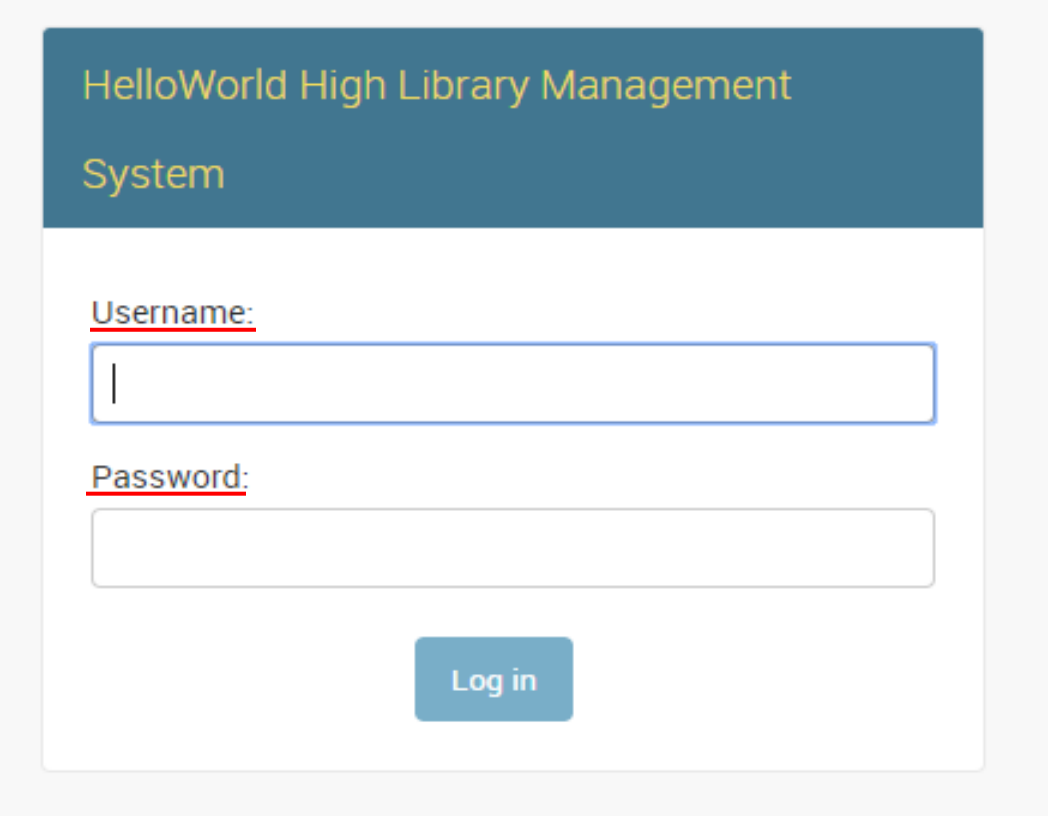
Who can access the Library System?



SUPERUSER:
The Superuser has the highest amount of access to the system and may perform modifications to the website as well as the database itself.

LIBRARIANS/FACULTY/TEACHERS:
The Librarians and faculty may retrieve records from the database, such as books and users, and modify/add/delete these records.

STUDENTS:
The students may **view** information pertaining to each book i.e. when it's due back to the library, who checked it out, etc.



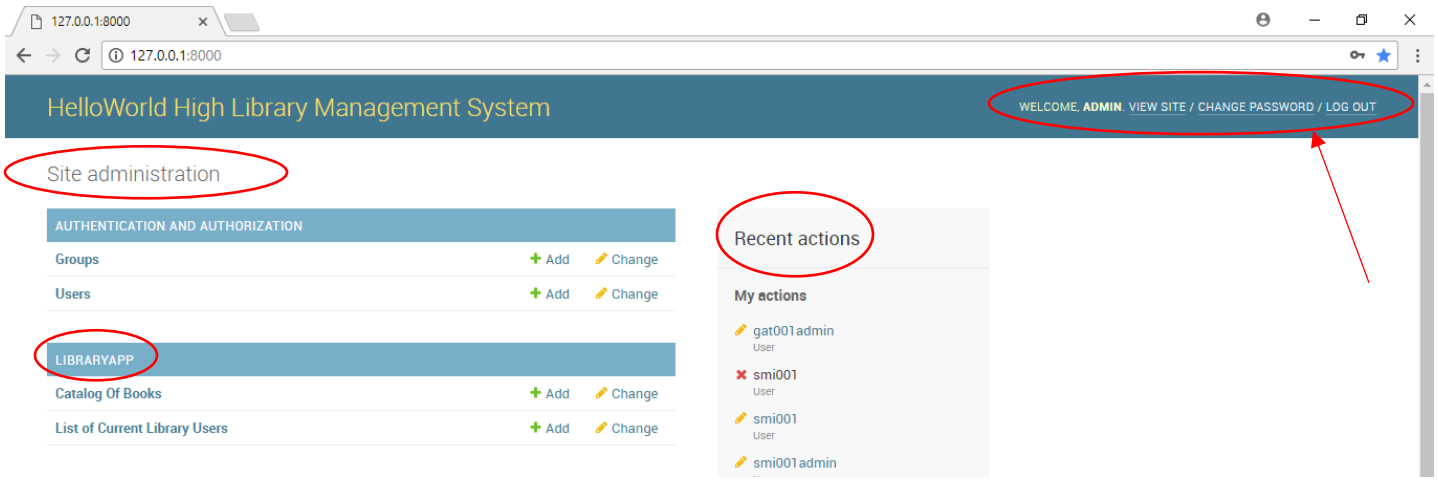
The Main Login Screen

(“HOMESCREEN” OF WEBSITE)

http://127.0.0.1:8000/

Main Menu

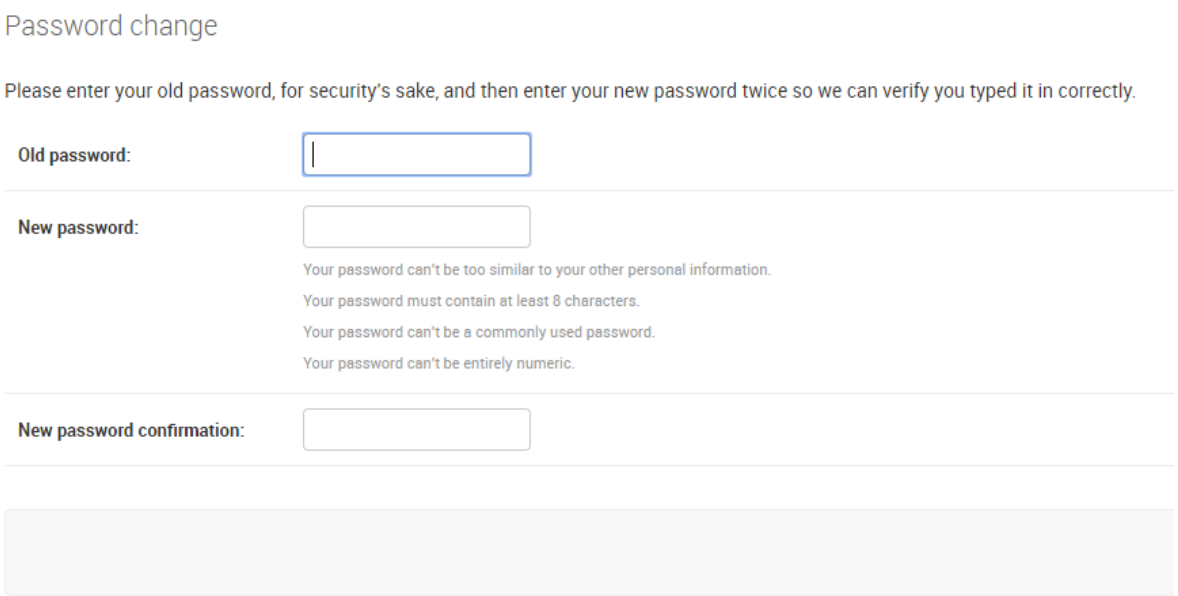
If you have logged in as a Library/Faculty user, you will arrive at the Main Menu of the library system.



The menu is separated into four major sections:

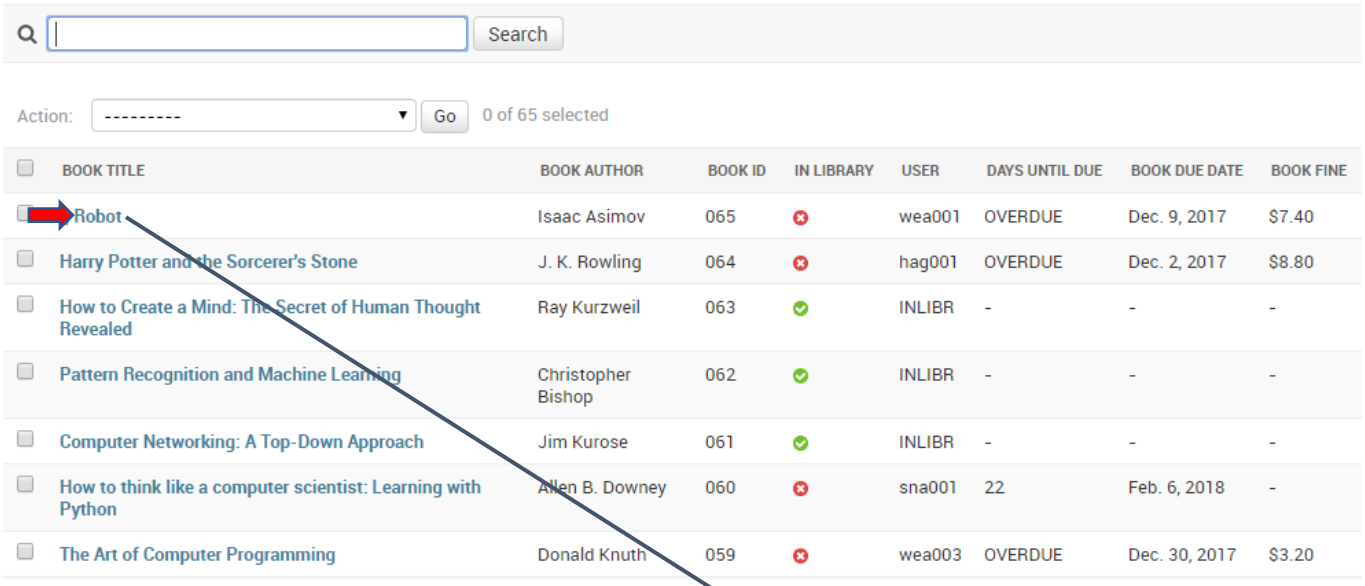
- *"Authentication and Authorization": Security Administration*
  - **"Groups":** Brings you to a page to manage the permissions of the two groups, Library Administrators and Library Users. Also allows for deleting/adding
  - **"Users":** Brings you to a page to manage the account information; CONTROLS USERS ACCESSING THE SITE, NOT CHECKING OUT BOOKS The same people can be found under the Library Application, but that list of people controls the checking out of books

- *Library Application:*
  - The core of the system, contains all library functionality
    - **Catalog of Books:** The list of books in library, containing all information of each book
    - **List of Current Library Users:** The list of users in system, containing all information of each person
- *Recent Actions:*
  - A List of all recent actions taken by the person currently logged in, such as modifications to a book or person
- *Sub-options (Located in top-right)*
  - View Site: When clicked at any page, redirects back to Main Menu
  - Change Password: Redirects user to a Change Password screen, asks for old password & new password
  - Log Out: Logs current user out of the Library System

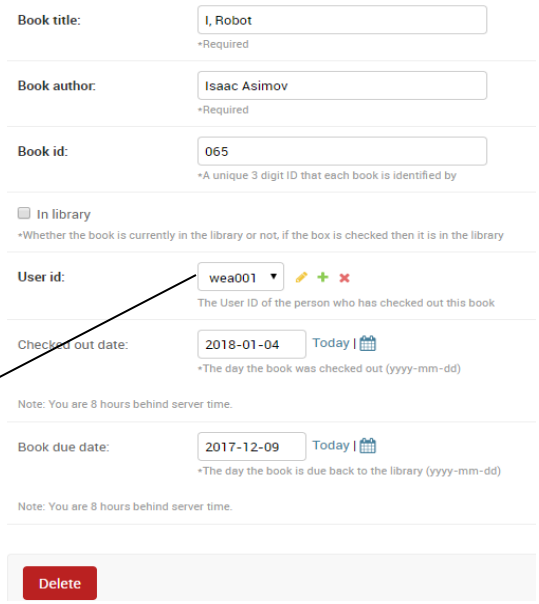


Password-Change Screen

Book Catalog:



The book catalog contains the list of books currently registered into the Library Management Database. Each record, (book), has 7 fields as shown to the right when the book is clicked.



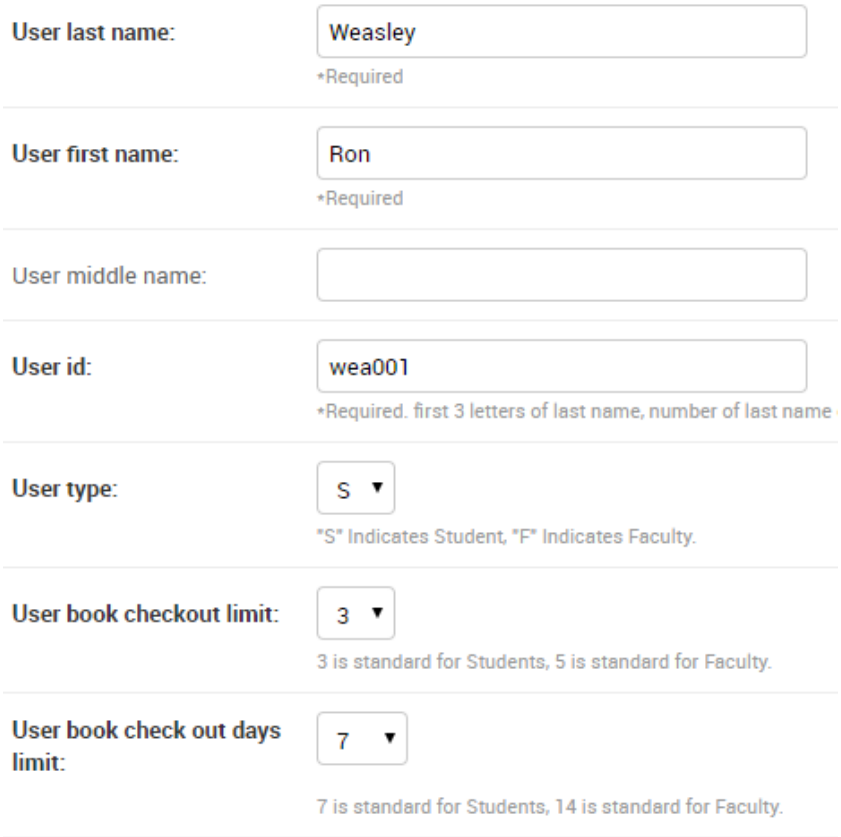
Linked directly to the User table, here you can access user's profiles

Screen shown once a book is clicked, editable

Library Users:

The List of Current Library Users is a list of all people stored to be connected to the books.

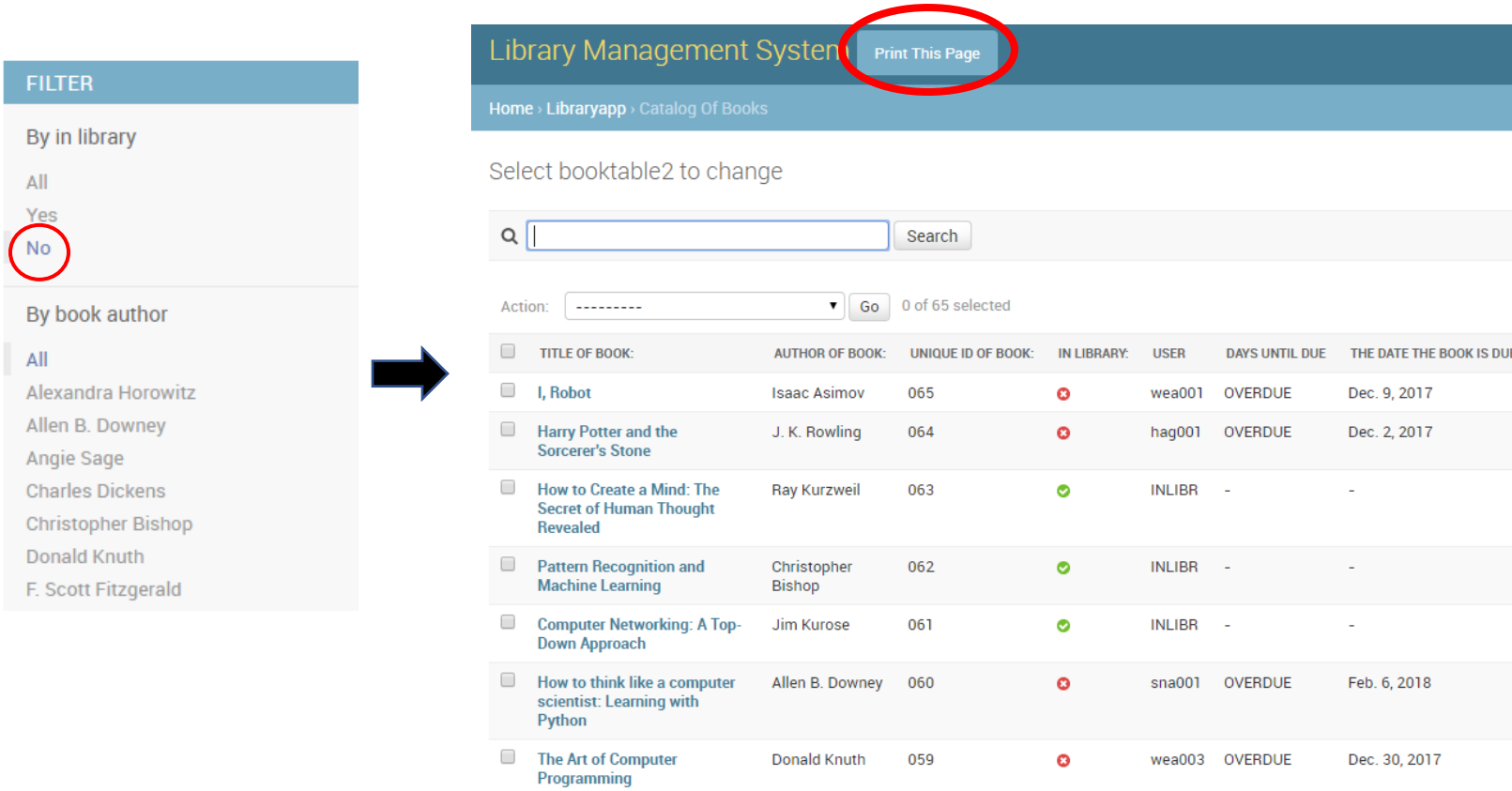
Similar to the Book Catalog, another editable screen of user information is brought up when a user is clicked.



User "profile"

Reports

In each section of the library, there is a "Filter" able to generate a current report from the database on demand.



By clicking the "No" link under "Filter" it generates a report of all checked out books, with the following information:

- Who Checked it out
- Whether the book is **overdue**
- Fine currently owed on the book. (The fine is automatically calculated by multiplying 20 cents by the number of days the book is overdue)

To print this report, click the "Print This Page" button at the top

Library Data Model

***PostgreSQL is the Database software used to store the library's information.**

USER_TABLE	-Table containing user info
user_last_name	-User's last name
user_first_name	-User's first name
user_middle_name	-User's middle name
user_id PRIMARY KEY	-6 Digit Unique user ID
user_type	-Whether user is Faculty/Teacher or student
user_book_checkout_limit	- # of books that can be checked out at once
user_book_check_out_days_limit	- # of days a book can be checked out for

BOOK_TABLE	-Table containing book info
book_title	-Title of book
book_author	-Author of book
book_id	-Unique 3-digit book ID number
user_id FOREIGN KEY	-ID of user that has book checked out
in_library	-Status of book; checked in or out
checked_out_date	-Date the book was checked out
book_due_date	-Date the book is due back to library
days_until_due	-Days until the book is due back

Development

The HelloWorld High Library Management System was created with the web-framework Django, using the languages Python and HTML. The Database software PostgreSQL is connected to the main project through a ***"settings.py"*** file. The tables with fields as well as logic for them to follow are created in the ***"models.py"*** (shown below)

All scripts can be found under the folders:

C:\EthanPrice_Ellensburg_WA_FBLA_Coding\django-apps\libraryproject2\libraryapp

And:

C:\EthanPrice_Ellensburg_WA_FBLA_Coding\django-apps\libraryproject2\libraryproject2

"models.py"

```
#Create the fields that will be used and displayed of the users: Title, author, id, if its in library, date it was checked out, when its due, days
class booktable2(models.Model):
    book_title = models.CharField("Title of Book:", max_length=100, null=False, help_text='*The Title of the book.')
    book_author = models.CharField("Author of Book:", max_length=40, null=False, help_text='*The Author of the book.')
    book_id = models.CharField("Unique ID of Book:", max_length=3, primary_key=True, help_text='*The unique ID number of the book. (From 001-999)'
    in_library = models.BooleanField("In Library:", null=False, default=True, help_text='*(When the checkbox is checked, the book is currently in
    #Foreign Key, makes this database relational as it connects to the User Table to get ID's
    user_id = models.ForeignKey(usertable2, on_delete=models.SET_NULL, null=True, default='INLIBR', help_text='The User ID of the person who has c
    checked_out_date = models.DateField("The date the Book was Checked Out:", blank=True, null=True, help_text="*yyy-mm-dd The date the book was c
    book_due_date = models.DateField("The Date the Book is Due Back:", blank=True, null=True, help_text='*yyyy-mm-dd The date the book is due back
    days_until_due = models.IntegerField("Days Until the Book is Due:", null=True, blank=True, editable=False, help_text='The number of days until

    #Function that calculates and returns the number of days until a book is due dependent on today's date and when it is due
    #Returns OVERDUE if overdue
    def calc_days_until_due(self):
        if self.book_due_date != None:
            if self.book_due_date > datetime.date.today():
                delta = self.book_due_date - datetime.date.today()
                return delta.days
            else:
                return "OVERDUE"

    #Function that calculates and returns the fine amount dependent on whether a book is due, and for how long it was due for.
    #Multiplies number of days it has been overdue by 0.2 to return a dollar amount, incrementing by 20 cents a day
    def calc_fine(self):
        if self.book_due_date != None:
            if self.book_due_date < datetime.date.today():
                calcdays = abs(self.book_due_date - datetime.date.today())
                calcfine = float(calcdays.days * 0.2)
                return "$" + str("%.2f" % round(calcfine, 2))

    def user(self):
        return self.user_id
```

Back to top

HELLOWORLDHIGH LIBRARY

MANAGEMENT SYSTEM USER GUIDE