

# Library Management System

## Problem Statement:

Create an online web application for the purpose of an efficient and effective Library Management System in any institute for registered students. (**ADD PAYMENT INTEGRATION**)

## Features of Project:

1. **User-friendly:**  
Our website can be used by both students (borrower) and the librarian. We have separate portals for both.
2. **Easy surfing:**  
Books can be searched based on author, book genre, details, etc.
3. **E-library concept:**  
Students can issue their books online without being physically present and the librarian can manage book records and student accounts online.

## Requirements:

1. Django version 3.2.3
2. Python 3.9.2
3. django-crispy-forms
4. MySQL

## Relational Schema:

In our project, we have implemented tables using models.  
We have the following tables:

### 1. GENRE:

This defines the book category.  
Attributes are:  
**name**- just to name the category.

### 2. LANGUAGE:

The language in which the book is published.  
Attributes are:  
**name**- the language name.

### 3. BOOK:

This relation contains information about a book.  
Attributes are:  
**title**- title of the book.  
**Author**- writer of the book content. Also a foreign key from "Author".  
**Summary**- brief description about the book.  
**Isbn**- unique identification code for a particular copy of a book.  
**Genre**- category.  
**Language**- It is the foreign key from "language" in models.

**total\_copies**- number of copies of a particular type of book which is owned by the library.  
**available\_copies**- number of copies of a particular type of book which are remaining un-issued.

#### **4. AUTHOR:**

Relation containing details of the author of the book.

Attributes are:

**first\_name**

**last\_name**

**date\_of\_birth**

**date\_of\_death**

#### **5. STUDENT:**

Relation containing details of students.

Attributes are:

**roll\_no**- this is chosen as the primary key.

**Name**

**branch**

**contact\_no**

**total\_books\_due**

#### **6. BORROWER:**

Relation containing borrowing details.

Attributes are:

**student**- foreign key from “student”.

**Book**- foreign key from “book”.

**issue\_date**

**return\_date**- date of book

### **Project Structure**

- Our main project folder is “Library\_management”.
- We have an app named “management”.
- Book records are located in management\_book
- author details located in management\_author
- forms.py contain model form-  
**BookForm, BorrowForm, StudentForm, RatingForm**

### **Functionalities**

We have used a single app in our project named ‘management’.It contains all the database,template and views.

**Functions in views.py :**

- **index** – renders the home page index.html
- **login\_view**- renders the login page
- **logout\_view**-redirects user to index.html when user log out
- **register\_view**- as the name suggest it renders the registration page ‘register.html’

- **BookListView**- list all the books available in library and renders 'book\_list.html'
- **BookDetailView**-accepts a parameter pk ie the id of book renders the detail of book to 'book\_detail.html'
- **Student\_BookListView**-render the books issued by the user.
- **BookCreate**-renders for adding a new book.
- **BookUpdate**- accepts a parameter pk ie the id of book and render update form for book.
- **BookDelete**-accepts a parameter pk i.e the id of the book superuser wants to delete.
- **Student\_request\_issue**-issues a book to a student if he hasn't issued more than 10 books.
- **StudentCreate**-renders a form for user creation
- **StudentUpdate**-accepts a parameter pk i.e the id of student renders a form for updating information of book
- **StudentDelete**- accepts a parameter pk i.e the id of student and delete the student record from database
- **StudentDetail**- accepts a parameter pk i.e the id of student and renders the information of student ret- accepts a parameter pk i.e the id of borrower object and deletes the entry from database.
- **RatingUpdate**- accepts a parameter pk i.e the id of Reviews object and render a form for updating it
- **RatingDelete**- accepts a parameter pk i.e the id of Reviews object and deletes the entry from database.

## Use of Middleware

We have used a middleware "GoogleSearch ". Whenever any syntax error in the program codes is witnessed during the execution of website, a new tab automatically opens with a google search of that error and giving the user a crystal clear idea about the error and how can the user fix it.

## Github Link

<https://github.com/gauravsamar/LibraryManagement>

## Output

Anonymous user login page

Documentation - Google Docs x Local Library x +

127.0.0.1:8000

Grand Library Books Admin Login

# Hello, AnonymousUser

A library is a collection of materials, books or media that are easily accessible for use and not just for display purposes. It is responsible for housing updated information in order to meet the user's needs on a daily basis. A Library provides physical (hard copies documents) or digital access (soft copies) materials, and may be a physical location or a virtual space, or both. A library's collection can include printed materials and other physical resources in many formats such as DVDs, CDs and Cassette as well as access to information, music or other content held on bibliographic databases.

[List of Books»](#)

## Django

Django is a Python-based free and open-source web framework that follows the model-template-views architectural pattern. It is maintained by the Django Software Foundation, an American independent organization established as a 501 non-profit.

[View details»](#)

## MySQL

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

[View details»](#)

## Crispy Forms

Django-crispy-forms is an application that helps to manage Django forms. It allows adjusting forms' properties (such as method, send button or CSS classes) on the backend without having to re-write them in the template.

[View details»](#)

Type here to search


ENG INTL 01:16 07-06-2021

## Login page

Documentation - Google Docs x Local Library x +

127.0.0.1:8000/accounts/login/

Grand Library Books Admin Login



Username:

Password:

[submit](#)

[Lost password?](#)

Type here to search

ENG INTL 01:17 07-06-2021

## Superuser login

The screenshot shows a web browser window with two tabs: 'Documentation - Google Docs' and 'Local Library'. The address bar shows '127.0.0.1:8000'. The application's navigation bar includes links for 'Grand Library', 'Books', 'Admin', 'Add Student', 'Add Book', and 'StudentList'. A user profile dropdown for 'gauravsamar' is on the right. The main content area features a large 'Hello, gauravsamar' greeting, a paragraph about libraries, and a 'List of Books»' button. Below this are three cards for 'Django', 'MySQL', and 'Crispy Forms', each with a brief description and a 'View details»' link. The Windows taskbar at the bottom shows the search bar and various application icons.

## List of Books

This screenshot shows the 'List of Books' page in the Grand Library application. The browser window has the address '127.0.0.1:8000/books/'. The navigation bar is consistent with the previous page. A search bar with the placeholder 'Enter book title' and a 'Search' button is at the top. The 'Book List' section displays three book covers: 'Harry Potter and the Cursed Child', 'Game of Thrones', and '3 Idiots'. Each cover has its title and author below it. The Windows taskbar is visible at the bottom.

## New Student

Documentation - Google Docs x Local Library x +

127.0.0.1:8000/student/create/

Grand Library Books Admin Add Student Add Book StudentList gauravsamar

Roll no\*

Name\*

Branch\*

Contact no\*

Total books due\*

0

Email\*

Pic

Choose File No file chosen

Type here to search

ENG 01:18 07-06-2021

## New Book

Documentation - Google Docs x Local Library x +

127.0.0.1:8000/book/create/

Grand Library Books Admin Add Student Add Book StudentList gauravsamar

Title\*

Author\*

Summary\*

Enter a brief description of the book

ISBN\*

13 Character ISBN number

Type here to search

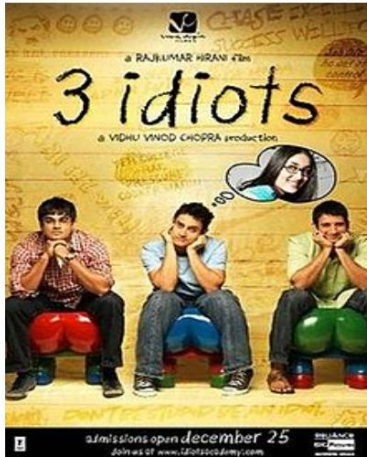
ENG 01:19 07-06-2021

## Issue page

Documentation - Google Docs x Local Library x +

127.0.0.1:8000/book/5

Grand Library Books Admin Add Student Add Book StudentList gauravsamar



**Title: 3 Idiots**

**Author:** Chetan Bhagat

**Summary:** In college, Farhan and Raju form a great bond with Rancho due to his refreshing outlook. Years later, a bet gives them a chance to look for their long-lost friend whose existence seems rather elusive.

**ISBN:** 11211U8219291

**Language:** English

**Genre:** FICTION

**Copies**

DELETE Update

no of avialable copies=19

Issue

**Gaurav** ★★★★★

A great book

Type here to search

ENG INTL 01:20 07-06-2021

## List of Issued Books


Documentation - Google Docs x Local Library x +

127.0.0.1:8000/student/book\_list

Grand Library Books Admin Add Student Add Book StudentList gauravsamar

Enter book title Search

### Book List



Game Of



3 Idiots

Type here to search


ENG INTL 01:23 07-06-2021

## Student Detail

Documentation - Google Docs x Local Library x +

127.0.0.1:8000/student/8

Grand Library Books Admin Add Student Add Book StudentList gauravsamar



**Name: Gaurav**

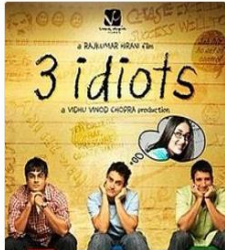
**Id:**123

**Contact:** 8769775701

**Branch:** CS

**Books due:** 1

DELETE Update



Type here to search

ENG INTL 01:21 07-06-2021

## Student List

Documentation - Google Docs x Local Library x +

127.0.0.1:8000/student/

Grand Library Books Admin Add Student Add Book StudentList gauravsamar

enter student Roll\_no or name or email Search

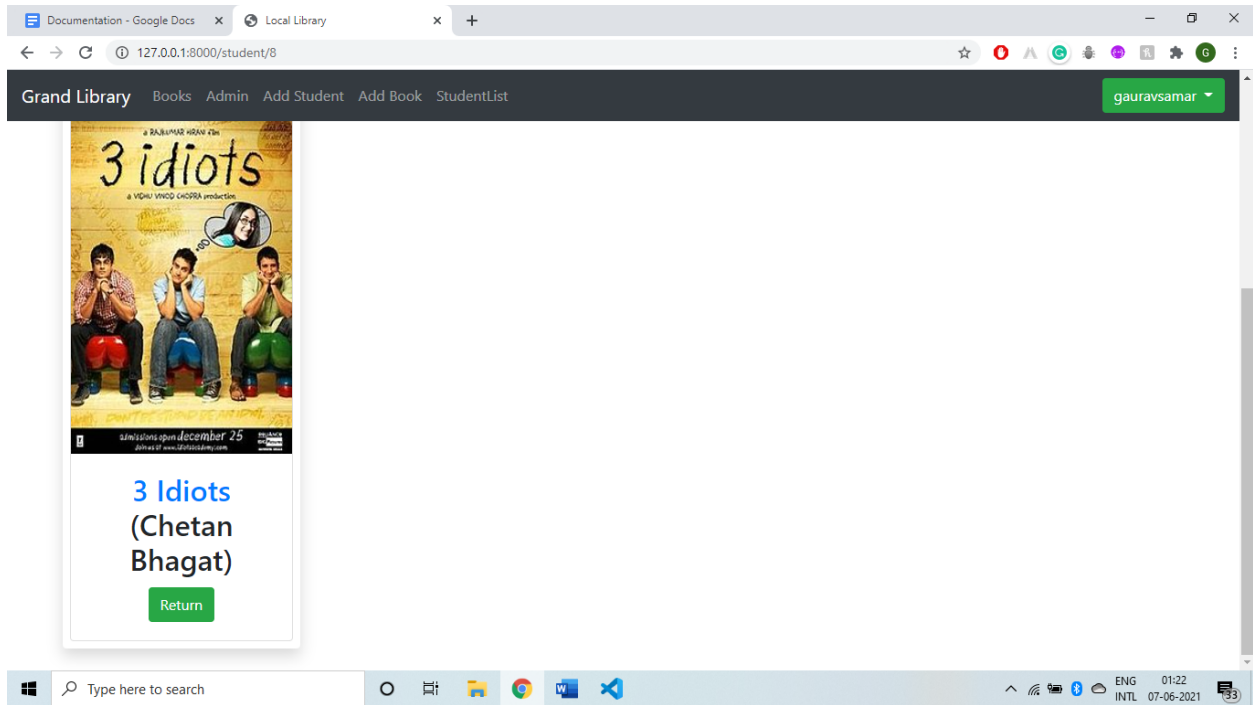
- 17075007
- 17075003
- 123

Type here to search

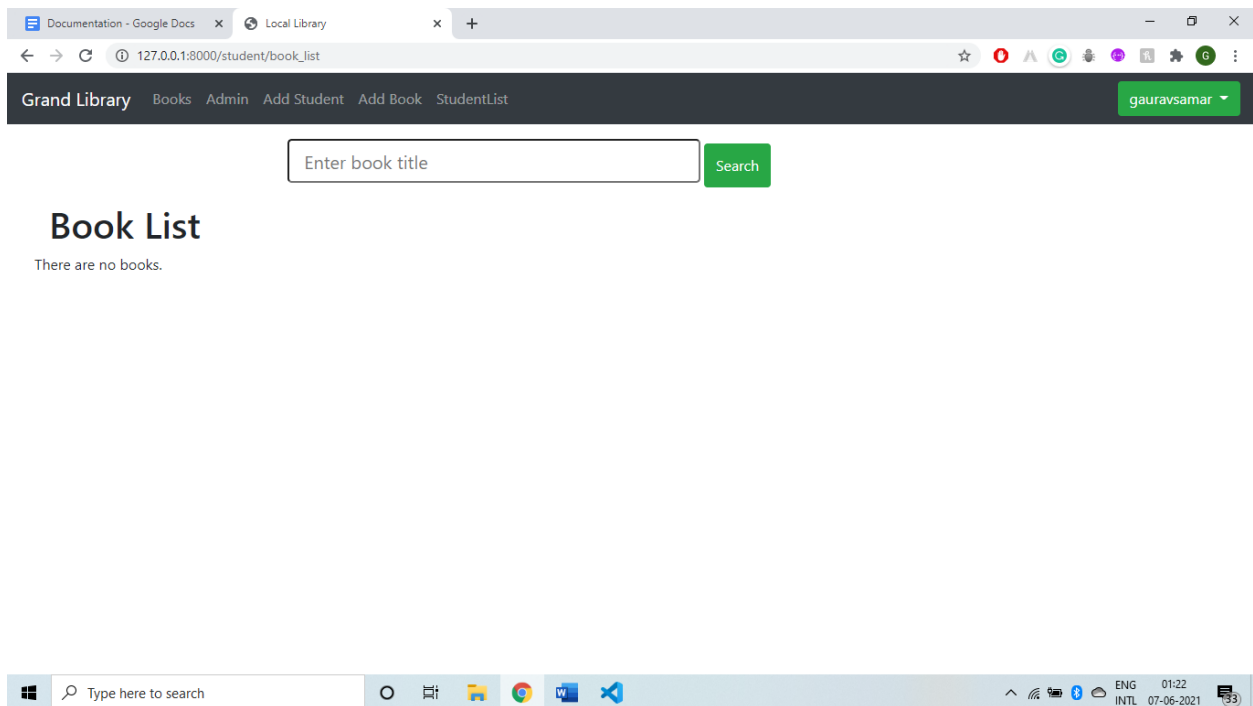
ENG INTL 01:20 07-06-2021

Return





## Return Success page



## **crispy\_forms**

**Django-crispy-forms** is an application that helps to manage **Django forms**. It allows adjusting **forms'** properties (such as method, send button or CSS classes) on the backend without having to re-write them in the template.