

# Library Assignment Report

10<sup>th</sup> August 2019

By- Vedansh Kedia  
Backend Machine Learning Intern  
CSE Department, IIT Bombay

## **Abstract**

Local Library Website developed facilitates the users with various online library features centrally. This product is built on Django 2 framework of Python for Back-End with HTML, CSS and Bootstrap features for the Front-End. This product was developed keeping in mind the current and technological prospects of Venter application. Due to non-conventional features of this product makes it unique and more fruitful for the user.

Sign Up to the website to use.

Link to Website: <https://vedanshlibrary.herokuapp.com>

Link to GitHub Repository: [https://github.com/VedanshKedia/Django\\_Library](https://github.com/VedanshKedia/Django_Library)

## Features

### Login/Sign Up:

These features registers and allows user to enter into the website to access the resources. It includes various Validations.

### Catalogue:

This feature displays the overview of the contents of the Library, i.e., it tells the user the count of all the resources available in the library. Such as:

- Books which are present in the library
- How many books are available for issue?
- How many authors are registered with our library?
- How many books are issued by the user?

### All Books:

A user can visit the page of All books from where user can see, add, update and delete the book, based on the permissions assigned.

- Only Site admin can add, update and delete the book from the website itself and not logging into the admin panel.
- Rest can only have the view access to the books.

### Borrowed book:

Borrowed booklist feature is divided into two categories of 'All Borrowed Books' and 'My Borrowed Books', where the 'All Borrowed' feature's access is given to librarian and the site admin, whereas 'my borrowed' feature is given to all the users.

Renew issued book feature

- This feature allows the Librarian and the Site admin to renew a particular book to a particular due back date.
- A reader does not have access to it.

### E Mail Issued Book:

This a non-conventional feature of the Library website which allows readers to get the issued book as an attachment in the email. This feature is accessible by all users.

### Search Widget:

Library Website has a very nicely implemented search widget where any user can search the Books and Authors from their respective lists.

- This search feature does not require the user to input the exact name of the book or author.
- It can search for the book or author by just receiving the keywords in any random order

## Concepts Learned and Implemented

New concepts Learned on implementation of Library assignment:

### Advanced Techniques to handle the admin page:

Listing the attributes which needs to displayed on the admin panel for a particular object for make display more informative.

- Add list filters, which allow the user to filter out the desired objects, based on the filter selected by the user.
- Add a field feature to the admin model which will control the fields displayed and laid out in forms on admin panel.
- Add sections to the admin panel

Class based views, which are the inbuilt generic classes of Django:

- *ListView*: Returns a list of objects in a particular model.
- *DetailView*: Returns information present in a particular object selected by the user by using its primary key
- *UpdateView*: It directly helps to render a form for updating a particular object of a model assigned in the class.
- *Create View*: It returns a form which can be rendered to create new object of a particular model.
- *Delete View*: It help in deleting a particular object from the model.

### Pagination:

Since, it is not a good practice to display all the objects in the list at one go, and make the page size large. Hence, Pagination feature allows us the break the list into several pages and display only the specified count in one page.

User authentication, roles and Permissions:

Permission: We can assign the permissions to users for tasks performed in the website. These permissions can be defined in the models and the views. Views and models containing the permission feature can be accessed by only user which has those permissions assigned.

Roles: There exist three predefined role in a website, regular, staff, admin.

- Staff access to view the admin panel and complete database, whereas
- Admin has access to perform all CRUD operations on them

### Get\_queryset:

This inbuilt function of class-based views returns the objects to the template which are then supposed to be rendered.

This feature helped to uniquely build the search widget feature. With certain tweaks to its conventional usage, I implemented the search widget centrally for all the views and applied the individual backend for every feature.

Tweaks removed the necessity of building a separate view and the form for the search widget.

### **Mixins:**

Mixins are the inbuilt classes of Django which help in assigning some features to the class-based Views. These are inherited by the view class for its properties. For example:

- *LoginRequiredMixin*: It imposes a compulsion for a view to be accessed only by the Logged in User.
- *UserPassesTestMixin*: It consist of an inbuilt function “*test\_func()*” which allows to add any custom conditions, which is then to be passed by the user to access a that particular view.

### **Automation Testing:**

Automation Testing helps to test the product programmatically, where the database is created before testing and then destroyed after the execution of the test class.

A separate class is created for every elementary feature of website and testcases are written as the functions, which needs to be passed by the website.

Learned how to:

- Initialize and define the database for testing
- Receive a GET response from the particular view
- Asserts which should be raised based on the type of response returned by the GET.

## Website Database Structure

### Models:

Book	Author	BookInstances	Genre	Language
Image	Image	ID	Name	Name
Title	First_Name	Book		
Author	Last_Name	Imprint		
Summary	Date_of_Birth	Due_Back		
ISBN	Date_of_Death	Borrower		
Genre		Status		

The above-mentioned table contains the Model Name with their corresponding attribute names declared and defined in the models of the website.

### Relationships:

Book-to-BookInstances:      One to Many

Book-to-Genre:              Many to Many

Book-to-Language:         Many to One

Author-to-Book:             One to Many

## References

1. [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial\\_local\\_library\\_website](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website)
2. <https://docs.djangoproject.com/en/2.1/topics/>
3. <https://devcenter.heroku.com/articles/git>
4. <https://coverage.readthedocs.io/en/latest/>