

Django Library Management System

1. Introduction:-

This is django based web application named as Library Management sytem.

Django Server: It is inbuilt server that is given by Django itself.it default run on port:8000.

Create Django Project- First we have install Django.

Then create django project using command- `django-admin startproject projectname`

Create Django App- jump on project folder create app using comand-
`python manag.py startapp appname`

I have created 1 project and 2 apps:

1: Library_Management_System:

files:

setting.py:

1. add apps in INSTALLED_APP
2. give mysql database conectivity
3. give a template base directory path

1: Book_Service:

files:

1 - models.py – It is basically Model. That cotains class

class name: Book

class variable field:

title: that contain title of a book.

Description:that contain description ofa book

2 - views.py – It is basically View we logic for data CRUD . That cotains functions

Function name: base

Argument:request object

Return:render on home page

Function name: booklist

Argument:request object

Return:render on booklist page and show model data

Function name: create_book

Argument:request object

Return:render on createbook page and show modeldata form for addbook

Function name: update_book

Argument:request object,id(int)

Return:render on updatebook page and show modeldata form for updatebook

Function name: delete_book

Argument:request object,id(int) //delete book by given id

Return:redirect to booklist page

3 - forms.py – It is basically Form. That cotains class

class name: BookForm

About: it create form for model Book

4 - urls.py

About: In this file I have given a url for HTML page.

1: Book_Service:

files:

1 - models.py – It is basically Model. That cotains class

class name: CustomerUser // create customusermodel for email login

class variable field:

Django Library Management System

email: Takes email as Field

password: Takes password as Field

- 2 - views.py – It is basically View we logic for data CRUD . That cotains functions

Function name: signup

Argument: request object

Return: render on signup page

- 3 - forms.py – It is basically Form. That cotains class
class name: CustomerUserForm

About: it create form for model CustomUser

- 4 - urls.py

About: In this file I have given a url for HTML page(Signup,Login Student).

Server Run Comand- python manage.py runserver

ADMIN-

we can create admin using command: python manage.py createsuperuser

1.admin signup: using email,password,conform password.

2.admin login:using email and password.