Report Title

Documentation of

Library Management System

(Keywordio)

Prasad Adhav | Assessment | 10/06/2022

**Database**: My-SQL

**Editor**: VS Code

**Front-End** **Technology**: Bootstrap 5.2

*This is the brief documentation of the app called “****Library Management System****”.*

## Views.py

1. home:

def home(request):

  return render(request,'home.html')

1. admin\_signup:

def admin\_signup(request):

    if request.method=="POST":

        fm=UserCreationForm(request.POST)

        # print(fm)

        if fm.is\_valid():

            uname=fm.cleaned\_data['username']

            upass=fm.cleaned\_data['password1']

            #print(uname)

            #print(upass)

            u=User.objects.create\_user(username=uname,password=upass,

is\_superuser=True,is\_staff=True)

            u.save()

            return HttpResponseRedirect('/user\_login/')

    else:

        fm=UserCreationForm()

    return render(request,'admin\_signup.html',{'form':fm})

# VIEW FOR ADMIN SIGNUP END

1. sign\_up:

def signup(request):

    if request.method=="POST":

        fm=UserCreationForm(request.POST)

        # print(fm)

        if fm.is\_valid():

            uname=fm.cleaned\_data['username']

            upass=fm.cleaned\_data['password1']

            #print(uname)

            #print(upass)

            u=User.objects.create\_user(username=uname,password=upass,

is\_staff=True)

            u.save()

            return HttpResponseRedirect('/user\_login/')

    else:

        fm=UserCreationForm()

    return render(request,'signup.html',{'form':fm})

1. user\_login:

# VIEW FOR USER LOGIN  START

def user\_login(request):

    if request.method=="POST":

        fm=AuthenticationForm(request=request,data=request.POST)

        if fm.is\_valid():

            uname=fm.cleaned\_data['username']

            upass=fm.cleaned\_data['password']

            u=authenticate(username=uname,password=upass)

            if u is not None:

                print("valid user")

              login(request,u)

                return HttpResponseRedirect('/dashboard/')

    else:

        fm=AuthenticationForm()

    return render(request,'user\_login.html',{'form':fm})

1. user\_logout:

# VIEW FOR USER LOGOUT START

def user\_logout(request):

    logout(request)

    return HttpResponseRedirect('/')

    #return render(request,'user\_logout.html')

# VIEW FOR USER LOGOUT END

1. dashboard:

def dashboard(request):

    #coolecting authenticated user is from the session

    c=request.user

    cuid=c.id

    # print(cuid)

    u=User.objects.get(id=cuid)

    print(u.is\_superuser)

    b=Book.objects.all()

    content={}

    content['data'] = b

    content['is\_superuser']=u.is\_superuser

    return render(request,'dashboard.html',content)

1. add\_book:

# VIEW FOR ADDING BOOK START

def add\_book(request):

    if request.method=="POST":

        bname=request.POST['bname']

        bauthor=request.POST['bauthor']

        # print(bname)

        # print(bauthor)

        b=Book.objects.create(name=bname,author=bauthor)

        b.save()

        return HttpResponseRedirect('/dashboard/')

    else:

        return render(request,'add\_book.html')

1. delete\_book:

# VIEW FOR DELETE BOOK START

def delete\_book(request,rid):

    b=Book.objects.filter(id=rid)

    b.delete()

return HttpResponseRedirect('/dashboard/')

1. edit\_book:

# VIEW FOR Edit BOOK START

def edit\_book(request,rid):

    if request.method=="POST":

        uname=request.POST['bname']

        uauthor=request.POST['bauthor']

        b=Book.objects.filter(id=rid)

        b.update(name=uname,author=uauthor)

        return HttpResponseRedirect('/dashboard/')

    else:

        b=Book.objects.filter(id=rid)

        content={}

        content['data']=b

        return render(request,'edit\_book.html',content)

urls.py

from django.contrib import admin

from django.urls import path

from libapp import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',views.home,name='home'),

    path('admin\_signup/',views.admin\_signup,name='admin\_signup'),

    path('signup/',views.signup,name='signup'),

    path('user\_login/',views.user\_login,name='user\_login'),

    path('user\_logout/',views.user\_logout,name='user\_logout'),

    path('dashboard/',views.dashboard,name='dashboard'),

    path('add\_book/',views.add\_book,name='add\_book'),

    path('delete/<rid>',views.delete\_book,name='delete\_book'),

    path('edit/<rid>',views.edit\_book,name='edit\_book'),

]

models.py

from django.db import models

# Create your models here.

class Book(models.Model):

    name=models.CharField(max\_length=50)

    author=models.CharField(max\_length=50)

settings.py

DataBase(MySQL)

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'mydatabase',

        'USER':'root',

        'HOST':'localhost',

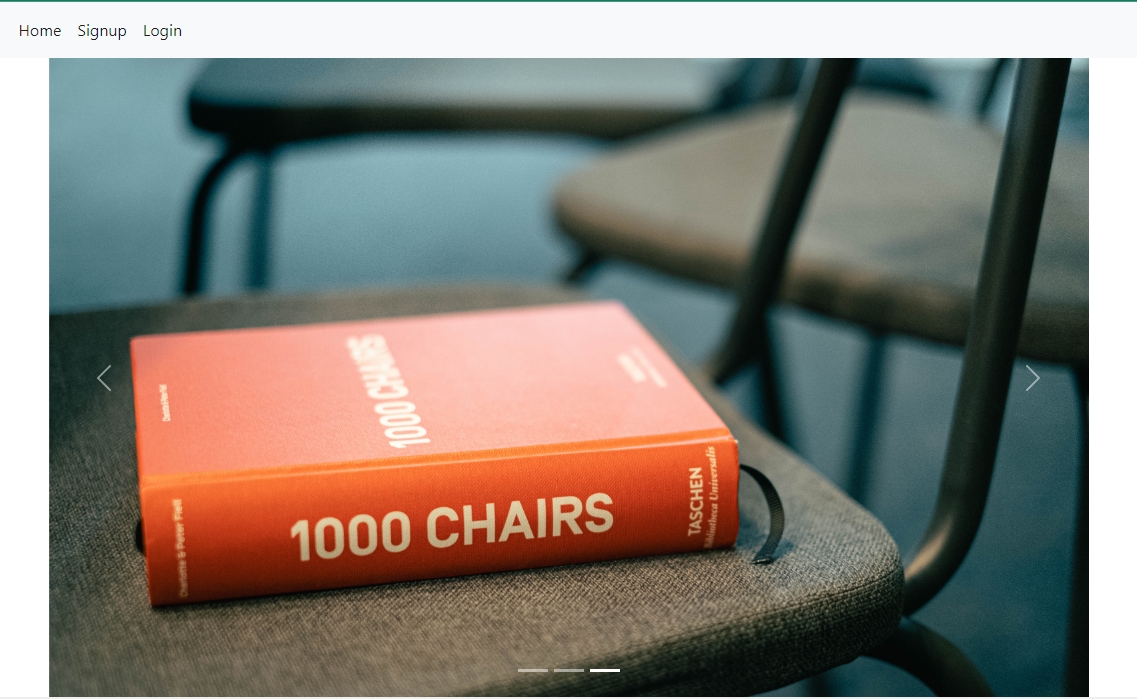
        'PASSWORD':''

               }

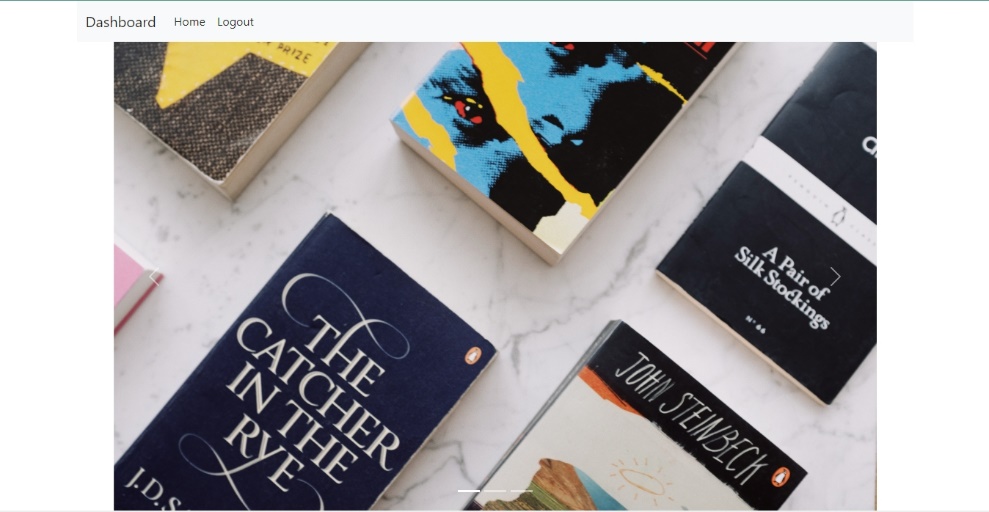
           }

Web View

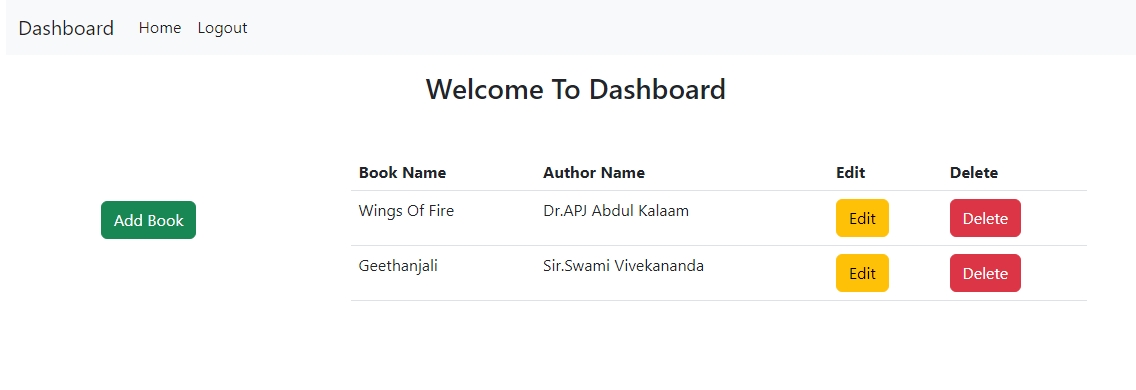
Home(before login)



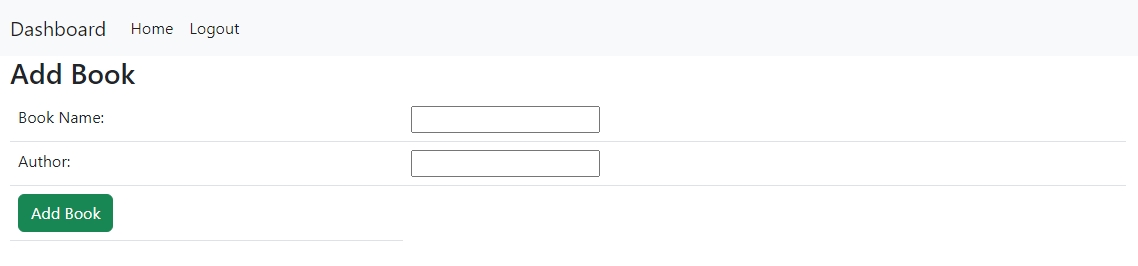
Home(after login)



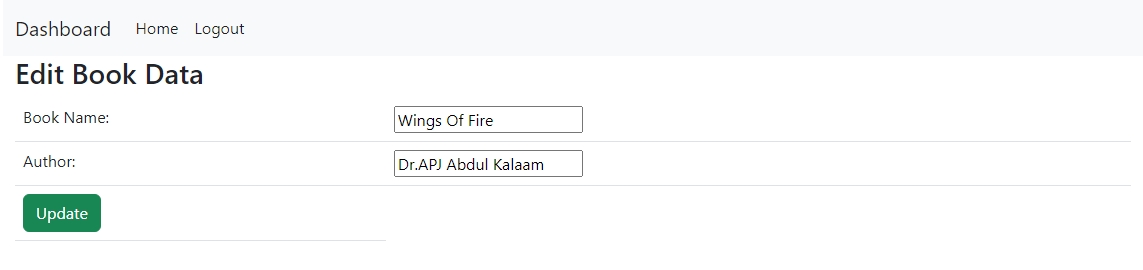
Dashboard



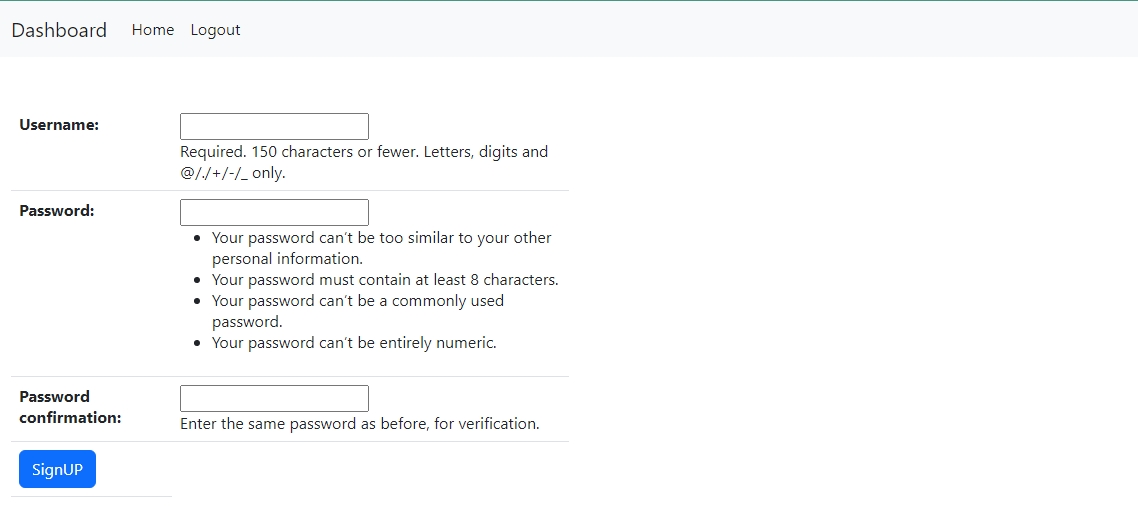
Add Book



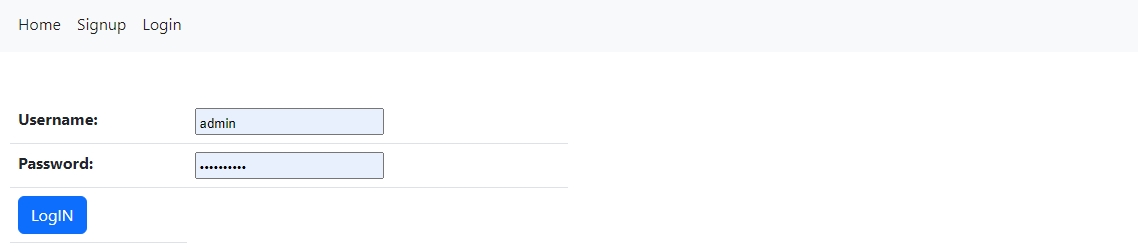
Update Book



Admin\_signup/student\_signup



Login



Inserted Records In DataBase

localhost/phpMyAdmin

