M.Tharun Chowdary
190031607

PFSD SKILLING 11

Library Management System

Week-11

Task1: Create Views, URL Mapping, Template System with necessary page views.

Task2: Create database and tables which are required for the project.

Views.py

```
from django.shortcuts import render

from django.http import HttpResponseRedirect

from . import forms,models

from django.http import HttpResponseRedirect

from django.contrib.auth.models import Group

from django.contrib import auth

from django.contrib.auth.decorators import login_required,user_passes_test

from datetime import datetime,timedelta,date

from django.core.mail import send_mail

from librarymanagement.settings import EMAIL_HOST_USER



def home_view(request):

    if request.user.is_authenticated:

        return HttpResponseRedirect('afterlogin')

    return render(request,'library/index.html')


#for showing signup/login button for student

def studentclick_view(request):

    if request.user.is_authenticated:

        return HttpResponseRedirect('afterlogin')

    return render(request,'library/studentclick.html')


#for showing signup/login button for teacher

def adminclick_view(request):

    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return render(request,'library/adminclick.html')
```

Urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/',include('django.contrib.auth.urls') ),
    path('', views.home_view),

    path('adminclick', views.adminclick_view),
    path('studentclick', views.studentclick_view),


    path('adminsignup', views.adminsignup_view),
    path('studentsignup', views.studentsignup_view),
    path('adminlogin', LoginView.as_view(template_name='library/adminlogin.html')),
    path('studentlogin',
LoginView.as_view(template_name='library/studentlogin.html')),

    path('logout', LogoutView.as_view(template_name='library/index.html')),
    path('afterlogin', views.afterlogin_view),

    path('addbook', views.addbook_view),
    path('viewbook', views.viewbook_view),
    path('issuebook', views.issuebook_view),
    path('viewissuedbook', views.viewissuedbook_view),
    path('viewstudent', views.viewstudent_view),
    path('viewissuedbookbystudent', views.viewissuedbookbystudent),

    path('aboutus', views.aboutus_view),
    path('contactus', views.contactus_view),

]
```

Database setup:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


class StudentExtra(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    enrollment = models.CharField(max_length=40)
    branch = models.CharField(max_length=40)
    #used in issue book
    def___str__(self):
        return self.user.first_name+'['+str(self.enrollment)+']'
    @property
    def get_name(self):
        return self.user.first_name
    @property
    def getuserid(self):
        return self.user.id



class Book(models.Model):
    catchoice= [
        ('education', 'Education'),
        ('entertainment', 'Entertainment'),
        ('comics', 'Comics'),
        ('biography', 'Biographie'),
        ('history', 'History'),
        ]
    name=models.CharField(max_length=30)
    isbn=models.PositiveIntegerField()
    author=models.CharField(max_length=40)
    category=models.CharField(max_length=30,choices=catchoice,default='education')
    def___str__(self):
        return str(self.name)+"["+str(self.isbn)+']'



def get_expiry():
    return datetime.today() + timedelta(days=15)
class IssuedBook(models.Model):
    #moved this in forms.py
    #enrollment=[(student.enrollment,str(student.get_name)+'
['+str(student.enrollment)+']') for student in StudentExtra.objects.all()]
    enrollment=models.CharField(max_length=30)
    #isbn=[(str(book.isbn),book.name+' ['+str(book.isbn)+']') for book in
Book.objects.all()]
    isbn=models.CharField(max_length=30)
    issuedate=models.DateField(auto_now=True)
    expirydate=models.DateField(default=get_expiry)
    def__str__(self):
        return self.enrollment
```