core\views.py

```python
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.contrib import messages

from .models import (
    User, StudentProfile, TeacherProfile, ClassRoom, Subject,
    Attendance, Result, Lecture, Assignment
)
from .forms import (
    UserRegisterForm, StudentProfileForm, TeacherProfileForm,
    ClassRoomForm, SubjectForm, AttendanceForm, ResultForm,
    LectureForm, AssignmentForm, AssignmentSubmissionForm,
    LeaveRequestForm, FeedbackForm
)

def home(request):
    return render(request, 'core/home.html')

def user_login(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        pwd = request.POST.get('password')
        user = authenticate(request, username=username, password=pwd)
        if user is not None:
            login(request, user)
            if user.is_admin():
                return redirect('admin_dashboard')
            elif user.is_teacher():
                return redirect('teacher_dashboard')
            else:
                return redirect('student_dashboard')
        else:
            messages.error(request, 'Invalid credentials')
    return render(request, 'core/login.html')

@login_required
def user_logout(request):
    logout(request)
    return redirect('login')

@login_required
def admin_dashboard(request):
    if not request.user.is_admin():
        return redirect('home')
    context = {
        'students_count': StudentProfile.objects.count(),
        'teachers_count': TeacherProfile.objects.count(),
        'classes_count': ClassRoom.objects.count(),
        'subjects_count': Subject.objects.count(),
    }
    return render(request, 'core/admin_dashboard.html', context)

@login_required
def teacher_dashboard(request):
    if not request.user.is_teacher():
        return redirect('home')
    teacher_profile = request.user.teacher_profile
    subjects = teacher_profile.subjects.all()
    assignments = Assignment.objects.filter(created_by=teacher_profile).order_by('-created_at')
    context = {
        'teacher': teacher_profile,
        'subjects': subjects,
        'assignments': assignments,
    }
    return render(request, 'core/teacher_dashboard.html', context)

@login_required
def student_dashboard(request):
    if not request.user.is_student():
        messages.error(request, "Access denied. You are not a student.")
        return redirect('home')

    if not hasattr(request.user, "student_profile"):
        messages.error(request, "Student profile does not exist. Contact admin.")
        return redirect('home')

    student_profile = request.user.student_profile
    subjects = Subject.objects.filter(classroom=student_profile.classroom)
    attendance = Attendance.objects.filter(student=student_profile)
    results = Result.objects.filter(student=student_profile)

    return render(request, 'core/student_dashboard.html', {
        'student': student_profile,
        'subjects': subjects,
        'attendance_list': attendance,
        'results': results,
    })

@login_required
def add_classroom(request):
    if not request.user.is_admin():
        return redirect('home')
    if request.method == 'POST':
        form = ClassRoomForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Classroom created.')
            return redirect('admin_dashboard')
    else:
        form = ClassRoomForm()
    return render(request, 'core/add_classroom.html', {'form': form})

@login_required
def add_subject(request):
    if not request.user.is_admin():
        return redirect('home')
    if request.method == 'POST':
        form = SubjectForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Subject created.')
            return redirect('admin_dashboard')
    else:
        form = SubjectForm()
    return render(request, 'core/add_subject.html', {'form': form})

@login_required
def add_teacher(request):
    if not request.user.is_admin():
        return redirect('home')
    if request.method == 'POST':
        user_form = UserRegisterForm(request.POST)
        profile_form = TeacherProfileForm(request.POST)
        if user_form.is_valid() and profile_form.is_valid():
            user = user_form.save(commit=False)
            user.role = 'TEACHER'
            user.save()
            teacher_profile = profile_form.save(commit=False)
            teacher_profile.user = user
            teacher_profile.save()
            messages.success(request, 'Teacher created.')
            return redirect('admin_dashboard')
    else:
        user_form = UserRegisterForm()
        profile_form = TeacherProfileForm()
    return render(request, 'core/add_teacher.html', {
        'user_form': user_form,
        'profile_form': profile_form
    })

@login_required
def add_student(request):
    if not request.user.is_admin():
        return redirect('home')
    if request.method == 'POST':
        user_form = UserRegisterForm(request.POST)
        profile_form = StudentProfileForm(request.POST)
        if user_form.is_valid() and profile_form.is_valid():
            user = user_form.save(commit=False)
            user.role = 'STUDENT'
            user.save()
            student_profile = profile_form.save(commit=False)
            student_profile.user = user
            student_profile.save()
            messages.success(request, 'Student created.')
            return redirect('admin_dashboard')
    else:
        user_form = UserRegisterForm()
        profile_form = StudentProfileForm()
    return render(request, 'core/add_student.html', {
        'user_form': user_form,
        'profile_form': profile_form
    })

@login_required
def mark_attendance(request):
    if not request.user.is_teacher():
        return redirect('home')
    if request.method == 'POST':
        form = AttendanceForm(request.POST)
        if form.is_valid():
            attendance = form.save(commit=False)
            attendance.marked_by = request.user.teacher_profile
            attendance.save()
            messages.success(request, 'Attendance marked.')
            return redirect('teacher_dashboard')
    else:
        form = AttendanceForm()
    return render(request, 'core/mark_attendance.html', {'form': form})

@login_required
def add_result(request):
    if not request.user.is_teacher():
        return redirect('home')
    if request.method == 'POST':
        form = ResultForm(request.POST)
        if form.is_valid():
            result = form.save(commit=False)
            result.created_by = request.user.teacher_profile
            result.save()
            messages.success(request, 'Result saved.')
            return redirect('teacher_dashboard')
    else:
        form = ResultForm()
    return render(request, 'core/add_result.html', {'form': form})

```

```python
@login_required
def upload_lecture(request):
    if not request.user.is_teacher():
        return redirect('home')
    if request.method == 'POST':
        form = LectureForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            messages.success(request, 'Lecture uploaded.')
            return redirect('teacher_dashboard')
    else:
        form = LectureForm()
    return render(request, 'core/upload_lecture.html', {'form': form})

@login_required
def upload_assignment(request):
    if not request.user.is_teacher():
        return redirect('home')
    if request.method == 'POST':
        form = AssignmentForm(request.POST, request.FILES)
        if form.is_valid():
            assignment = form.save(commit=False)
            assignment.created_by = request.user.teacher_profile
            assignment.save()
            messages.success(request, 'Assignment uploaded.')
            return redirect('teacher_dashboard')
    else:
        form = AssignmentForm()
    return render(request, 'core/upload_assignment.html', {'form': form})

@login_required
def submit_assignment(request, assignment_id):
    if not request.user.is_student():
        return redirect('home')
    assignment = get_object_or_404(Assignment, id=assignment_id)
    if request.method == 'POST':
        form = AssignmentSubmissionForm(request.POST, request.FILES)
        if form.is_valid():
            submission = form.save(commit=False)
            submission.assignment = assignment
            submission.student = request.user.student_profile
            submission.save()
            messages.success(request, 'Assignment submitted.')
            return redirect('student_dashboard')
    else:
        form = AssignmentSubmissionForm()
    return render(request, 'core/submit_assignment.html', {
        'form': form,
        'assignment': assignment
    })
```

```python

@login_required
def request_leave(request):
    if request.user.is_student():
        requester_type = 'STUDENT'
    elif request.user.is_teacher():
        requester_type = 'TEACHER'
    else:
        return redirect('home')

    if request.method == 'POST':
        form = LeaveRequestForm(request.POST)
        if form.is_valid():
            leave = form.save(commit=False)
            leave.requester = request.user
            leave.requester_type = requester_type
            leave.save()
            messages.success(request, 'Leave request submitted.')
            return redirect('student_dashboard' if requester_type == 'STUDENT' else 'teacher_dashboard')
    else:
        form = LeaveRequestForm()
    return render(request, 'core/request_leave.html', {'form': form})

@login_required
def send_feedback(request):
    if not (request.user.is_student() or request.user.is_teacher()):
        return redirect('home')
    if request.method == 'POST':
        form = FeedbackForm(request.POST)
        if form.is_valid():
            fb = form.save(commit=False)
            fb.sender = request.user
            fb.save()
            messages.success(request, 'Feedback sent to HOD.')
            if request.user.is_student():
                return redirect('student_dashboard')
            else:
                return redirect('teacher_dashboard')
    else:
        form = FeedbackForm()
    return render(request, 'core/send_feedback.html', {'form': form})

```