def search\_datatable(request):

# รับค่าพารามิเตอร์จากคำขอ GET

client\_id = request.GET.get('client\_id')

start\_date\_period = request.GET.get('start\_date\_period')

to\_date\_period = request.GET.get('to\_date\_period')

category\_id = request.GET.get('category\_id')

type\_id = request.GET.get('type\_id')

deadline\_days = request.GET.get('deadline\_days')

per\_month = request.GET.get('per\_month')

# กรองข้อมูล engagements ตามเงื่อนไขที่รับมาจากแบบฟอร์ม

engagements = Engagement.objects.all().order\_by('job\_code')

if client\_id:

engagements = engagements.filter(client\_id=client\_id)

if start\_date\_period:

engagements = engagements.filter(start\_date\_period\_\_gte=start\_date\_period)

if to\_date\_period:

engagements = engagements.filter(end\_date\_period\_\_lte=to\_date\_period)

if category\_id:

engagements = engagements.filter(category\_id=category\_id)

if type\_id:

engagements = engagements.filter(type\_id=type\_id)

if deadline\_days:

today = timezone.now().date()

engagements\_with\_deadlines = EngagementDetail.objects.filter(deadline\_\_gt=today, deadline\_\_lte=F('engagement\_\_end\_date\_period')).values('engagement\_id')

engagements = engagements.filter(id\_\_in=Subquery(engagements\_with\_deadlines))

if per\_month: # เพิ่มเงื่อนไขสำหรับกรองตามเดือนปัจจุบัน

current\_month = datetime.now().month

current\_year = datetime.now().year

engagements = engagements.filter(start\_date\_period\_\_month=current\_month, start\_date\_period\_\_year=current\_year)

engagement\_data = []

for engagement in engagements:

engagement\_detail = EngagementDetail.objects.filter(engagement=engagement)

remaining\_days = []

for detail in engagement\_detail:

difference = detail.deadline - date.today()

remaining\_days.append(difference.days)

engagement\_data.append({

'job\_code': engagement.job\_code,

'client\_id': engagement.client\_id,

'client\_\_company\_name': engagement.client.company\_name,

'start\_date\_period': engagement.start\_date\_period,

'administrator\_\_first\_name': engagement.administrator.first\_name,

'administrator\_\_last\_name': engagement.administrator.last\_name,

'type\_\_name\_th': engagement.type.name\_th if engagement.type else None,

'engagement\_detail\_deadlines': list(engagement\_detail.values('id', 'deadline')),

'remaining\_days': remaining\_days,

})

return JsonResponse(engagement\_data, safe=False)

start\_date = request.GET.get('start\_date')

end\_date = request.GET.get('end\_date')

administrator = request.GET.get('administrator')

# Filter engagement details based on start date and end date

filtered\_engagement\_details = EngagementDetail.objects.filter(deadline\_\_range=[start\_date, end\_date])