Views.py:  
def create\_internship\_calendar(request):  
 if request.method == 'POST':  
 form = InternCalendarForm(request.POST)  
 if form.is\_valid():  
 calendar = form.save(commit=False)  
 calendar.user = request.user  
 calendar.save()  
 # Calculate the number of days between start and end months  
 num\_days = (calendar.end\_month - calendar.start\_month).days + 1  
 # Create submission bins  
 for i in range(num\_days):  
 report = DailyAccomplishmentReport(internship\_calendar=calendar, order=i+1)  
 report.save()  
 return redirect('calendar\_detail', pk=calendar.pk)  
 else:  
 form = InternCalendarForm()  
 return render(request, 'internship\_calendar/set\_up\_calendar.html', {'form': form})  
  
def calendar\_detail(request, pk):  
 calendar = InternCalendar.objects.get(pk=pk)  
 reports = DailyAccomplishmentReport.objects.filter(internship\_calendar=calendar)  
  
 for report in reports:  
 report.calculated\_date = calendar.start\_month + timedelta(days=report.order - 1)  
  
 return render(request, 'internship\_calendar/calendar\_detail.html', {'calendar': calendar, 'reports': reports})  
  
def create\_daily\_report(request, calendar\_pk):  
 calendar = InternCalendar.objects.get(pk=calendar\_pk)  
 if request.method == 'POST':  
 form = DailyAccomplishmentReportForm(request.POST, request.FILES)  
 if form.is\_valid():  
 report = form.save(commit=False)  
 report.internship\_calendar = calendar  
 report.save()  
 return redirect('calendar\_detail', pk=calendar.pk)  
  
 else:  
 form = DailyAccomplishmentReportForm()  
 return render(request, 'internship\_calendar/create\_daily\_report.html', {'form': form, 'calendar': calendar})

Models.py:  
class InternCalendar(models.Model):  
 user = models.ForeignKey(intern, on\_delete=models.CASCADE)  
 start\_month = models.DateField()  
 end\_month = models.DateField()  
  
class DailyAccomplishmentReport(models.Model):  
 internship\_calendar = models.ForeignKey(InternCalendar, on\_delete=models.CASCADE)  
 date = models.DateField(default=now)  
 order = models.PositiveIntegerField(default=0)  
 text\_report = models.TextField(blank=True, null=True)  
 document\_report = models.FileField(upload\_to='documents/', blank=True, null=True)

Urls.py

path('create-calendar/', views.create\_internship\_calendar, name='set\_up\_calendar'),  
path('calendar/<int:pk>/', views.calendar\_detail, name='calendar\_detail'),  
path('calendar/<int:calendar\_pk>/create-daily-report/', views.create\_daily\_report, name='create\_daily\_report'),

forms.py:

class InternCalendarForm(forms.ModelForm):  
 class Meta:  
 model = InternCalendar  
 fields = ['start\_month', 'end\_month']  
  
class DailyAccomplishmentReportForm(forms.ModelForm):  
 class Meta:  
 model = DailyAccomplishmentReport  
 fields = ['date', 'text\_report', 'document\_report']

Internship\_Calendar

Calendar\_detail.html:  
{% extends "base.html" %}  
{% block content %}  
 <h2>Internship Calendar</h2>  
 <p>Start Month: {{ calendar.start\_month }}</p>  
 <p>End Month: {{ calendar.end\_month }}</p>  
 <h3>Daily Accomplishment Reports</h3>  
 <ul>  
 {% for report in reports %}  
  
 <li>  
 Date: {{ report.calculated\_date|date:"Y-m-d" }}  
 <a href="{% url 'create\_daily\_report' calendar.pk %}">Add Daily Report</a>  
  
 <p>{{ report.text\_report }}</p>  
 </li>  
 {% empty %}  
 <li>No reports available.</li>  
 {% endfor %}  
 </ul>  
  
{% endblock %}

Create\_daily\_report.html:  
{% extends "base.html" %}  
{% block content %}  
 <h2>Create Daily Accomplishment Report for {{ selected\_date|date:"Y-m-d" }}</h2>  
 <form method="post" enctype="multipart/form-data">  
 {% csrf\_token %}  
 {{ form.as\_p }}  
 <button type="submit">Submit</button>  
 </form>  
 <a href="{% url 'calendar\_detail' calendar.pk %}">Back to Calendar</a>  
{% endblock %}

set\_up\_calendar.html:

{% extends "base.html" %}  
{% block content %}  
 <h2>Create Internship Calendar</h2>  
 <form method="post" enctype="multipart/form-data">  
 {% csrf\_token %}  
 {{ form.as\_p }}  
 <button type="submit">Submit</button>  
 </form>  
{% endblock %}