**To Create Blog   
*1. Create Project***

***2. Create Apps Blog***

***In Module:***

**from** django.db **import** models  
**from** django.utils **import** timezone  
**from** django.contrib.auth.models **import** User  
  
  
  
  
**class** Post(models.Model):  
 STATUS\_CHOICES = (  
 (**'draft'**, **'Draft'**),  
 (**'published'**, **'Published'**),  
 )  
 title = models.CharField(max\_length=250)  
 slug = models.SlugField(max\_length=250, unique\_for\_date=**'publish'**)  
 author = models.ForeignKey(User, related\_name=**'blog\_posts'**)  
 body = models.TextField()  
 publish = models.DateTimeField(default=timezone.now)  
 created = models.DateTimeField(auto\_now\_add=True)  
 updated = models.DateTimeField(auto\_now=True)  
 status = models.CharField(max\_length=10, choices=STATUS\_CHOICES, default=**'draft'**)  
  
**class** Meta:  
 ordering = (**'-publish'**,)  
  
 **def** \_\_str\_\_(self):  
 **return** self.title

***3. Installed Apps: Enter : Blog***

***4. e:\python manage.py makemigrations blog***

***E:\>python manage.py sqlmigrate blog 0001***

***E:\>python manage.py migrate***

**Note : makemigrations** basically generates the SQL commands for preinstalled apps (which can be viewed in installed apps in settings.py) and your newly created apps' model which you add in installed apps.**It does not execute those commands in your database file.** So tables doesn't created after makemigrations.

After applying makemigrations you can see those SQL commands with **sqlmigrate** which shows all the SQL commands which has been generated by makemigrations.

**migrate** executes those SQL commands in database file.So after executing migrate all the tables of your installed apps are created in your database file.

***5. Add In administrative site :***

**from** .models **import** Post

admin.site.register(Post)

Run server…..

6. Admin:

**from** django.contrib **import** admin  
**from** .models **import** Post  
  
  
**class** PostAdmin(admin.ModelAdmin):  
 list\_display = (**'title'**, **'slug'**, **'author'**, **'publish'**, **'status'**)  
 list\_filter = (**'status'**, **'created'**, **'publish'**, **'author'**)  
 search\_fields = (**'title'**, **'body'**)  
 prepopulated\_fields = {**'slug'**: (**'title'**,)}  
 raw\_id\_fields = (**'author'**,)  
 date\_hierarchy = **'publish'** ordering = [**'status'**, **'publish'**]  
admin.site.register(Post, PostAdmin)

***Building List and Details View:***

**from** django.shortcuts **import** render, get\_object\_or\_404  
**from** .models **import** Post  
  
**def** post\_list(request, category=None):  
 posts = Post.published.all()

**return** render(request, **'blog/post/list.html'**, { **'posts'**: posts})  
 **def** post\_detail(request, year, month, day, post):  
 post = get\_object\_or\_404(Post, slug=post,  
 status=**'published'**,  
 publish\_\_year=year,  
 publish\_\_month=month,  
 publish\_\_day=day)  
 **return** render(request, **'blog/post/detail.html'**, {**'post'**: post})

**Urls.py:**

**from** django.conf.urls **import** url  
**from** . **import** views  
  
  
urlpatterns = [  
 *# post views  
 #url(r'^$', views.post\_list, name='post\_list'),* url(**r'^$'**, views.PostListView.as\_view(), name=**'post\_list'**),  
 url(**r'^(?P<year>\d{4})/(?P<month>\d{2})/(?P<day>\d{2})/(?P<post>[-\w]+)/$'**,  
 views.post\_detail,  
 name=**'post\_detail'**),  
]

Urls.py(Project)

**from** django.conf.urls **import** include, url  
**from** django.contrib **import** admin  
urlpatterns = [  
 url(**r'^admin/'**, include(admin.site.urls)),  
 url(**r'^blog/'**, include(**'blog.urls'**, namespace=**'blog'**, app\_name=**'blog'**)),  
]

**Modules.py**

**from** django.db **import** models  
**from** django.utils **import** timezone  
**from** django.contrib.auth.models **import** User  
**from** django.core.urlresolvers **import** reverse  
  
  
**class** PublishedManager(models.Manager):  
 **def** get\_queryset(self):  
 **return** super(PublishedManager, self).get\_queryset().filter(status=**'published'**)  
  
  
**class** Post(models.Model):  
 STATUS\_CHOICES = (  
 (**'draft'**, **'Draft'**),  
 (**'published'**, **'Published'**),  
 )  
 title = models.CharField(max\_length=250)  
 slug = models.SlugField(max\_length=250, unique\_for\_date=**'publish'**)  
 author = models.ForeignKey(User, related\_name=**'blog\_posts'**)  
 body = models.TextField()  
 publish = models.DateTimeField(default=timezone.now)  
 created = models.DateTimeField(auto\_now\_add=True)  
 updated = models.DateTimeField(auto\_now=True)  
 status = models.CharField(max\_length=10, choices=STATUS\_CHOICES, default=**'draft'**)  
  
 objects = models.Manager() *# The default manager.* published = PublishedManager() *# The Dahl-specific manager.* **class** Meta:  
 ordering = (**'-publish'**,)  
  
 **def** \_\_str\_\_(self):  
 **return** self.title  
  
 **def** get\_absolute\_url(self):  
 **return** reverse(**'blog:post\_detail'**, args=[self.publish.year,  
 self.publish.strftime(**'%m'**),  
 self.publish.strftime(**'%d'**),  
 self.slug])

**Templates**

**|--blog (base.html)**

{% load staticfiles %}  
<!DOCTYPE **html**>  
<**html**>  
<**head**>  
 <**title**>{% block title %}{% endblock %}</**title**>  
 <**link href="{% static "css**/blog.css" %}" rel="stylesheet">  
</**head**>  
<**body**>  
 <**div id="content"**>  
 {% block content %}  
 {% endblock %}  
 </**div**>  
 <**div id="sidebar"**>  
 <**h2**>My blog</**h2**>  
 <**p**>This is my blog.</**p**>  
 </**div**>  
</**body**>  
</**html**>

**Blog-post🡪details and list**

**Detaisl.html**

{% extends "blog/base.html" %}  
  
{% block title %}{{ post.title }}{% endblock %}  
  
{% block content %}  
 <**h1**>{{ post.title }}</**h1**>  
 <**p class="date"**>Published {{ post.publish }} by {{ post.author }}</**p**>  
 {{ post.body|linebreaks }}  
{% endblock %}

List.html

{% extends "blog/base.html" %}  
  
{% block title %}My Blog{% endblock %}  
  
{% block content %}  
 <**h1**>My Blog</**h1**>  
 {% for post in posts %}  
 <**h2**><**a href="{{ post.get\_absolute\_url }}"**>{{ post.title }}</**a**></**h2**>  
 <**p class="date"**>Published {{ post.publish }} by {{ post.author }}</**p**>  
 {{ post.body|truncatewords:30|linebreaks }}  
 {% endfor %}  
   
 {% include "pagination.html" with page=page\_obj %}  
{% endblock %}

Static—css-blog.css---

body {   
 margin:0;  
 padding:0;  
 font-family:helvetica, sans-serif;   
}  
  
a {   
 color:#00abff;  
 text-decoration:none;   
}  
  
h1 {   
 font-weight:normal;  
 border-bottom:1px solid #bbb;  
 padding:0 0 10px 0;  
}  
  
h2 {  
 font-weight:normal;  
 margin:30px 0 0;  
}  
  
#content {   
 float:left;  
 width:60%;  
 padding:0 0 0 30px;   
}  
  
#sidebar {   
 float:right;  
 width:30%;  
 padding:10px;  
 background:#efefef;   
 height:100%;  
}  
  
p.date {   
 color:#ccc;  
 font-family: georgia, serif;  
 font-size: 12px;  
 font-style: italic;   
}  
  
*/\* pagination \*/*.pagination {   
 margin:40px 0;   
 font-weight:bold;  
}  
  
*/\* forms \*/*label {   
 float:left;  
 clear:both;  
 color:#333;  
 margin-bottom:4px;   
}  
input, textarea {   
 clear:both;  
 float:left;  
 margin:0 0 10px;  
 background:#ededed;  
 border:0;  
 padding:6px 10px;  
 font-size:12px;  
}  
input[type=submit] {  
 font-weight:bold;  
 background:#00abff;  
 color:#fff;  
 padding:10px 20px;  
 font-size:14px;  
 text-transform:uppercase;   
}  
.errorlist {   
 color:#cc0033;  
 float:left;  
 clear:both;  
 padding-left:10px;   
}  
  
*/\* comments \*/*.comment {  
 padding:10px;  
}  
.comment:nth-child(even) {  
 background:#efefef;  
}  
.comment .info {  
 font-weight:bold;  
 font-size:12px;  
 color:#666;  
}