

SPRINT - 1

Urls.py :

```
urlpatterns = [  
    path("", views.college, name="college"),  
    path("notice/<int:myid>/", views.notice, name="notice"),  
    path("application_form/", views.application_form, name="application_form"),  
    path("edit_application/", views.edit_application, name="edit_application"),  
    path("status/", views.status, name="status"),  
    # Authentication  
    path("register/", views.register, name="register"),  
    path("login/", views.loggedin, name="login"),  
    path("logout/", views.loggedout, name="logout"),  
    # Admin  
    path("handle_admin/", views.handle_admin, name="handle_admin"),  
    path("users/", views.users, name="users"),  
    path("student_application/<int:myid>/", views.student_application, name="student_application"),  
    path("application_status/<int:pk>/", UpdatePostView.as_view(), name="application_status"),  
    path("approved_applications/", views.approved_applications, name="approved_applications"),  
    path("pending_applications/", views.pending_applications, name="pending_applications"),  
    path("rejected_applications/", views.rejected_applications, name="rejected_applications"),  
]
```

Code Explanation:

It is considered to be a good practice to create a separate urls file for each app. The urls are into three parts

- 1) For users
- 2) User Authentication
- 3) For admin

Models.py :

```
from django.db import models

from django.contrib.auth.models import User

from django.utils.timezone import now

from django.urls import reverse


class Application(models.Model):

    COURSES = (

        ('Computer Science Engineering', 'Computer Science Engineering'),

        ('Information Technology Engineering', 'Information Technology Engineering'),

        ('Electronics and Telecommunication Engineering', 'Electronics and Telecommunication Engineering'),

        ('Electronics Engineering', 'Electronics Engineering'),

    )

    STATUS = (

        ('Approved', 'Approved'),

        ('Pending', 'Pending'),

        ('Rejected', 'Rejected'),

    )

    user = models.OneToOneField(User, on_delete=models.CASCADE, blank=True, null=True)

    course = models.CharField(max_length=100, choices= COURSES)

    name = models.CharField(max_length=200)

    email = models.CharField(max_length=200)

    phone_no = models.CharField(max_length=200)

    address = models.TextField(max_length=200)

    student_profile = models.ImageField(upload_to="images")

    ssc_percentage = models.DecimalField(max_digits=4, decimal_places=2, null=True)
```

```
ssc_marksheet = models.ImageField(upload_to="images", null=True)
ssc_passing_certificate = models.ImageField(upload_to="images", null=True)
ssc_leaving_certificate = models.ImageField(upload_to="images", null=True)
hsc_percentage = models.DecimalField(max_digits=4, decimal_places=2, null=True)
hsc_marksheet = models.ImageField(upload_to="images", null=True)
hsc_passing_certificate = models.ImageField(upload_to="images", null=True)
hsc_leaving_certificate = models.ImageField(upload_to="images", null=True)
cet_percentile = models.DecimalField(max_digits=5, decimal_places=3, null=True)
cet_scorecard = models.ImageField(upload_to="images", null=True)
jee_percentile = models.DecimalField(max_digits=5, decimal_places=3, null=True)
jee_scorecard = models.ImageField(upload_to="images", null=True)
Application_Status = models.TextField(max_length=100, choices=STATUS, default="Pending")
message = models.TextField(max_length=100, default="")
```

```
def str(self):
    return self.name
```

```
def get_absolute_url(self):
    return reverse('users')
```

```
class Notice(models.Model):
    title = models.CharField(max_length=200)
```

```
def str(self):
    return self.title
```

```
class Detail(models.Model):
    title = models.ForeignKey(Notice, on_delete=models.CASCADE)
    notice = models.CharField(max_length=200)
```

```
def str(self):  
    return self.notice
```

Code Explanation:

The most important model of python college admission system is the Application model. It stores all the details of the students personal and educational details. The student while filling the application form gives all these details. The status and message are edited by the admin. Notice and Detail model stores the notice for first, second, third, and fourth year students. It is possible to add any notice for any category of students.

1. For the home page, all the notice for different year students will be shown (college.html):

```
<div class="container mt-4">  
    <h1>Important Notice</h1>  
<div class="row mt-4">  
    {% for i in notice %}  
        <div class="col-sm-6">  
            <div class="card">  
                <div class="card-body">  
                    <h5 class="card-title">{{i.title}}</h5>  
                    <p class="card-text"><a href="/notice/{{i.id}}/">View all recent updates.</a></p>  
                </div>  
            </div>  
        </div>  
    {% endfor %}  
</div>  
</div>
```

Views.py :

```
def college(request):  
  
    notice = Notice.objects.all()  
  
    return render(request, "college.html", {'notice':notice})
```

Code Explanation:

On the first page of the project all the notices will be displayed by using the for loop from the Notice model. Students can see the notice by clicking on the title regarding their year or branch.