

<http://127.0.0.1:8000/admin/>

页面中，保存有每个人的信息，可以随时进行删除，但是不能修改。

Admin Panel

Student Information:

Name	Class Name	Student ID	Selected Courses	Action
张三	一班	000001	• 生物	<button>Delete</button>
李四	二班	000002	• 生物	<button>Delete</button>
张三	二班	000003	• 地理	<button>Delete</button>
王五	一班	000004	• 地理	<button>Delete</button>

接下来先进行注册。

Student Registration

Name:

Class:

Register

Student Registration

Name:

Class:

Register

Registered Successfully

Your Student ID: 000006

Your Name: 王二麻子

[Login](#)

注册成功后会有提示，显示学号和一个登录的跳转界面。

Login

Student ID:

Name:

Login

为了防止重名，以及登录错误，选择同时验证学号和姓名的方式进行选课。

Welcome, 王二麻子

[Select Course](#)

[My Selected Courses](#)

点击选课后，可以看到一个下拉菜单。菜单内的内容为 Course 中的内容，为 shell 中手动添加的。

Select Course

Select Course:

生物 ▼

Select

生物
地理
历史

点击选课后，显示选课信息。

My Selected Courses

Selected Course: 生物

如再次点击选课，则无法选课。

Select Course

您已经选择了课程：生物。

亦可手动查询选课信息。

My Selected Courses

Selected Course: 生物

返回管理页面，可以看到新的学生信息。

王二麻子	计算机一班	000006	• 生物	Delete
------	-------	--------	------	--------

以下为项目视图。

```
urls.py x
1 from django.urls import path
2 from selection.views import *
3
4 urlpatterns = [
5     path('register/', register_student, name='register_student'),
6     path('login/', login, name='login'),
7     path('welcome/<str:student_id>/', welcome, name='welcome'),
8     path('select_course/<str:student_id>/', select_course, name='select_course'),
9     path('selected_courses/<str:student_id>/', selected_courses, name='selected_courses'),
10    path('admin/', admin, name='admin_panel'),
11    path('delete_student/<int:student_id>/', delete_student, name='delete_student'),
12 ]
13
```

```
urls.py x  models.py x
1 from django.db import models
2
3 class Course(models.Model): # 课程模型。
4     name = models.CharField(max_length=100)
5
6     def __str__(self):
7         return self.name
8
9 class Student(models.Model): # 学生模型。学生具有姓名、班级名称和学生ID属性。
10    name = models.CharField(max_length=100)
11    class_name = models.CharField(max_length=100)
12    student_id = models.CharField(max_length=6, unique=True)
13
14    def __str__(self):
15        return self.name
16
17 class Enrollment(models.Model): # 选课模型。关联了学生和课程。
18    student = models.OneToOneField(Student, on_delete=models.CASCADE)
19    course = models.ForeignKey(Course, on_delete=models.CASCADE)
20
21    def __str__(self):
22        return f"{self.student.name} - {self.course.name}"
23
```

views.py 内有注释说明，在此不做详细解释。