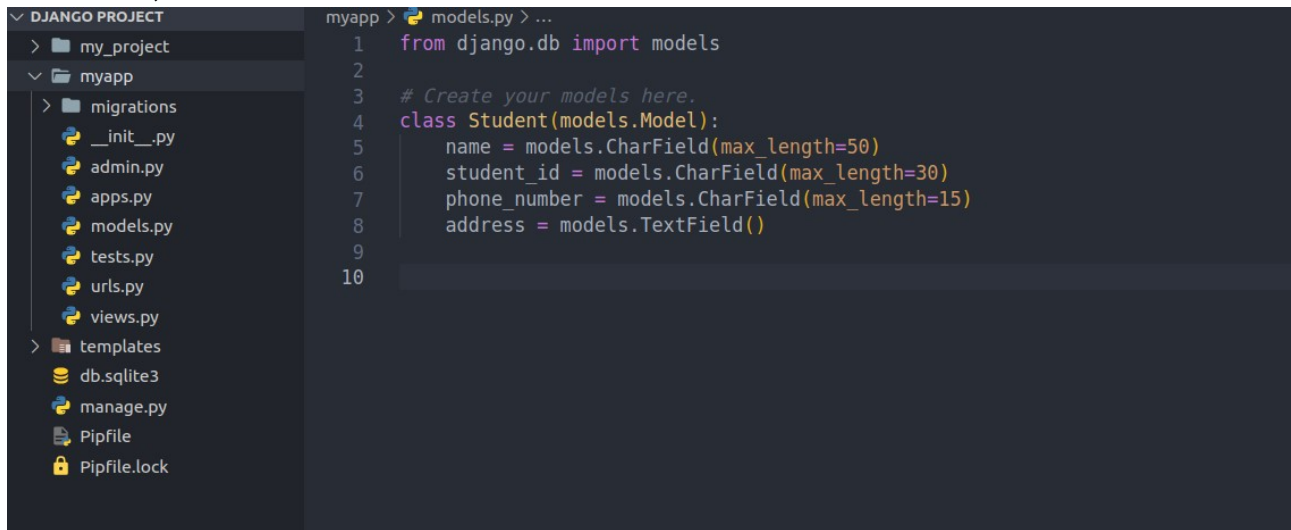


## Django Docs Part 02

### Model

মডেল হচ্ছে আমাদের Database এর টেবিল।



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the Django project structure: DJANGO PROJECT > my\_project > myapp. The myapp folder contains migrations, \_\_init\_\_.py, admin.py, apps.py, models.py, tests.py, urls.py, and views.py. The code editor shows the models.py file with the following code:

```
1 from django.db import models
2
3 # Create your models here.
4 class Student(models.Model):
5     name = models.CharField(max_length=50)
6     student_id = models.CharField(max_length=30)
7     phone_number = models.CharField(max_length=15)
8     address = models.TextField()
9
10
```

আমরা আমাদের অ্যাপ ফোল্ডারের models.py ফাইল এ নতুন একটি মডেল তৈরি করবো।

আমাদের হিসেব করতে হবে ( টাকা পয়সা ) এমন সব field বাদে বাকি field আমরা CharField ব্যবহার করতে পারি।

আমাদের যদি বেশি কিছু লেখার দরকার পরে তখন আমরা TextField ব্যবহার করবো।

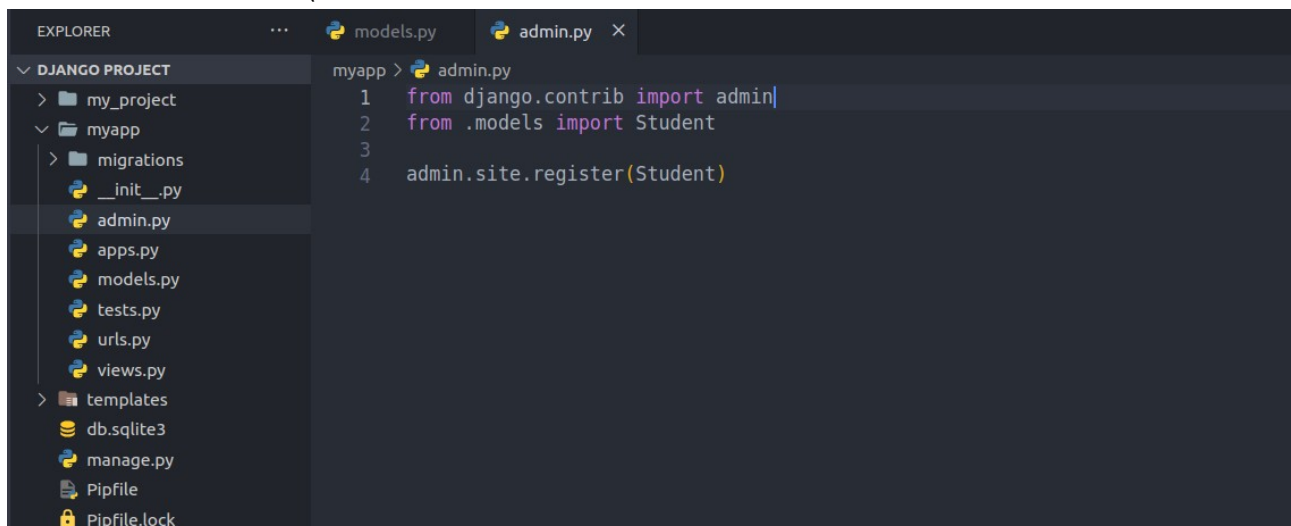
এছাড়া Django তে আরও অনেক field আছে।

<https://www.geeksforgeeks.org/django-model-data-types-and-fields-list/>

এই লিঙ্ক থেকে আমরা বাকিসব field গুলর কাজ দেখতে পারি।

আচ্ছা আমাদের মডেল বানানো শেষ।

এখন আমাদের এই মডেল এডমিন প্যানেলে দাখানর জন্য আমাদের এই মডেলটিকে admin.py ফিলের মধ্যে রেজিস্টার করতে হবে।



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the Django project structure: DJANGO PROJECT > my\_project > myapp. The myapp folder contains migrations, \_\_init\_\_.py, admin.py, apps.py, models.py, tests.py, urls.py, and views.py. The code editor shows the admin.py file with the following code:

```
1 from django.contrib import admin
2 from .models import Student
3
4 admin.site.register(Student)
```

এখানে আমরা নতুন ২ টি লাইন এড করবো।

from . Models import Student

admin.site.register(Student)

প্রতিবার নতুন মডেল বানানোর পরে আমরা এখানে প্রথমে মডেলটি import করবো তারপরে register করবো। এখানের কাজ আপাততো এতটুকুই।

আমরা প্রতিবার মডেল update কিংবা create করার পরে একবার এই কমান্ড ২ টা চালাবো।

```
python manage.py makemigrations
```

```
python manage.py migrate
```

এখন আমরা আমাদের সার্ভার রান করবো।

```
python manage.py runserver
```

আগের পর্বে আমরা superuser create করেছিলাম।

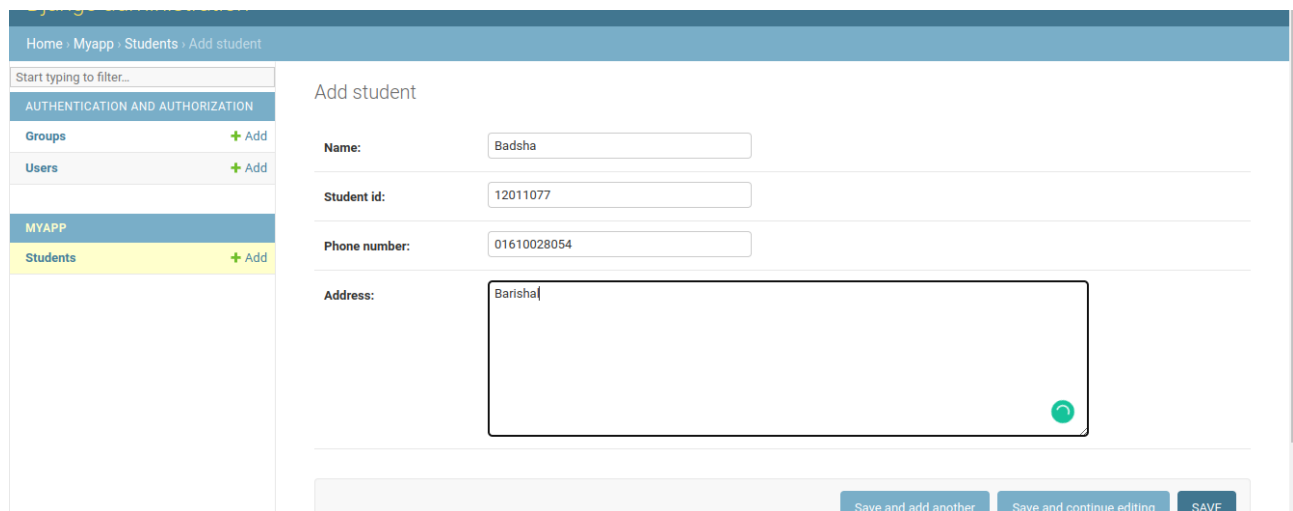
এখন আমরা আমাদের browser এ 127.0.0.1:8000/admin এই url এ যাবো।

লগইন করার পরে আমাদের সামনে

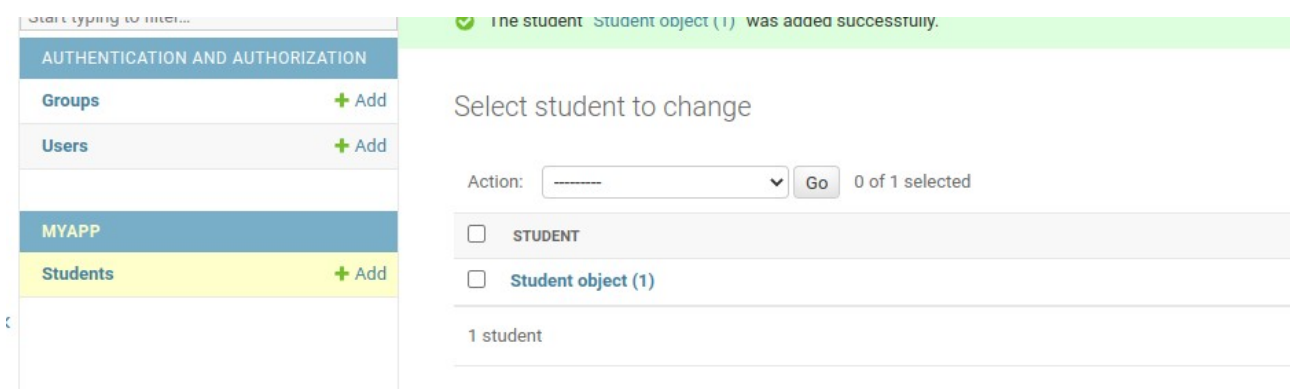


মডেলটি দেখাবে।

মডেলের ভিতরে ঢুকে আমরা একটা নতুন object তৈরি করি।

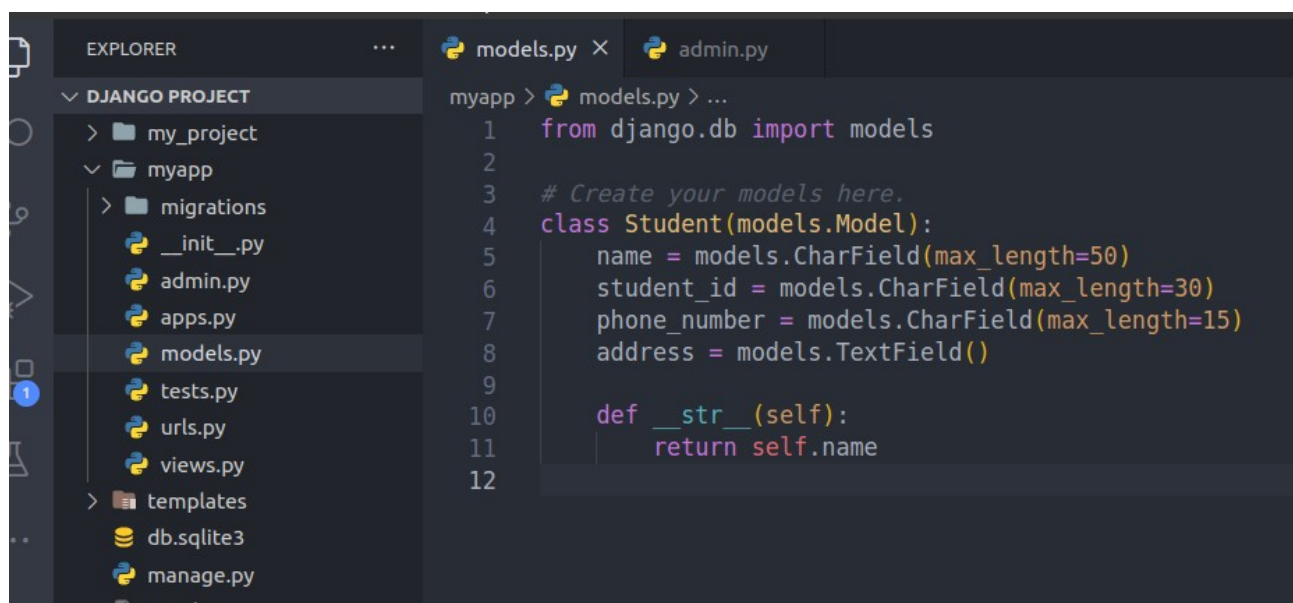


এখান থেকে আমরা মডেল CREATE, READ, DELETE, UPDATE সব কিছু করতে পারবো।



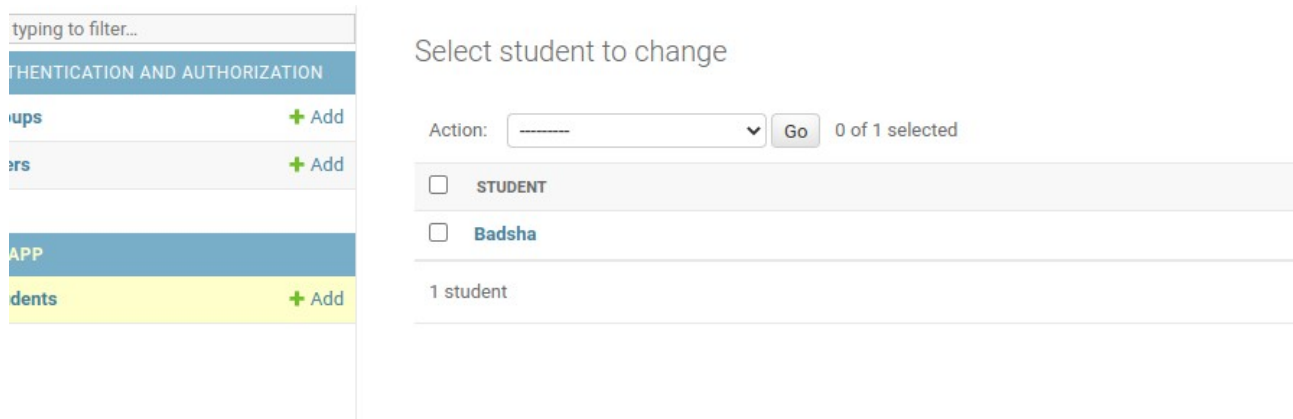
আমরা এখানে একজন স্টুডেন্টের object তৈরি করেছি। এরকম আরও অনেক object এখানে তৈরি হবে। এখন আমরা কিন্তু এই object দেখে বুঝতে পারতেছি না এইটা কোন স্টুডেন্টের। আমরা চাই এখানে এমন কিছু থাকবে যা দেখলে আমি বুঝতে পারবো এটা কার।

এজন্য আমাদের আবার models.py ফাইল যেতে হবে।



এখানে আমরা magic method টা ব্যবহার করবো।

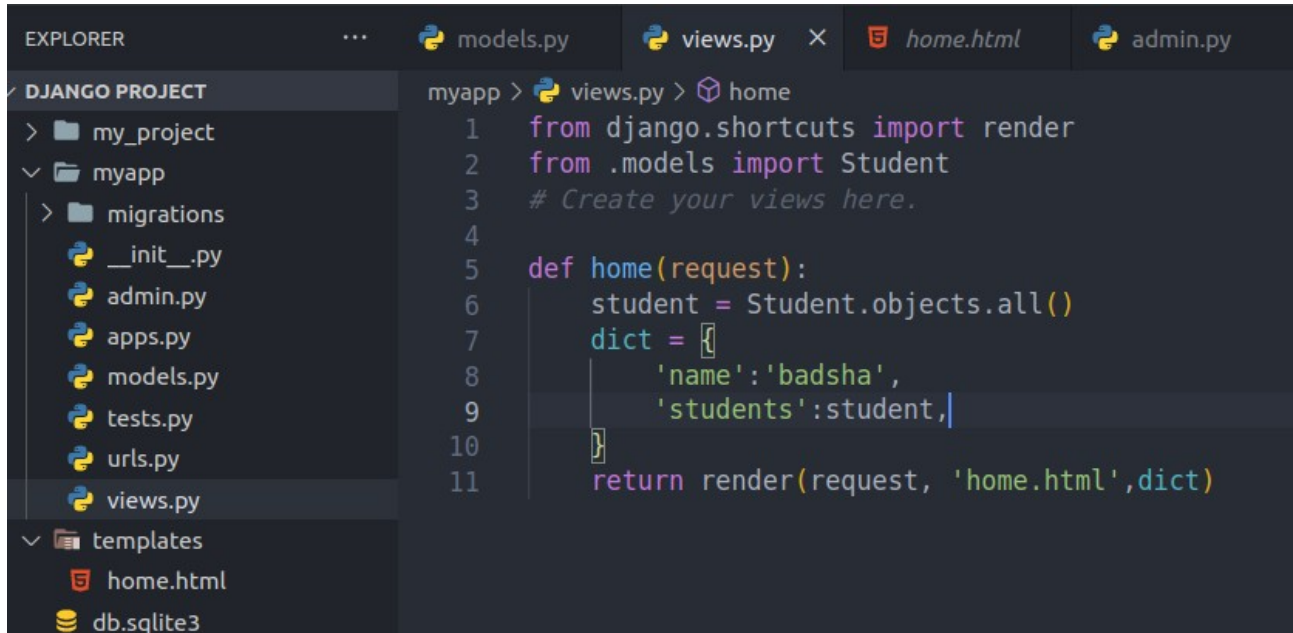
এখন যদি আমরা আমাদের এডমিন সাইট এ যাই



তাহলে এরকম দেখতে পারবো। যা দেখে আমরা সহজেই বুঝতে পারবো এটা কোন স্টুডেন্টের।

method add করার ক্ষেত্রে migration চালানোর দরকার পরে না কিন্তু চালালে কোনও অসুবিধা নাই।

এখন আমরা দেখব কিভাবে আমাদের মডেল এর এই ডাটা আমরা আমাদের template এ পাঠাতে পারি।



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a Django project structure with a 'myapp' folder containing 'migrations', '\_\_init\_\_.py', 'admin.py', 'apps.py', 'models.py', 'tests.py', 'urls.py', and 'views.py'. The 'templates' folder contains 'home.html' and 'db.sqlite3'. The code editor shows the 'views.py' file with the following code:

```
myapp > views.py > home
1 from django.shortcuts import render
2 from .models import Student
3 # Create your views here.
4
5 def home(request):
6     student = Student.objects.all()
7     dict = {}
8     'name': 'badsha',
9     'students': student,
10
11     return render(request, 'home.html', dict)
```

প্রথমেই আমি আমার views.py ফাইলটিতে গেলাম।

সেখানে আমি models থেকে Student মডেলটি import করলাম।

এরপর আমি Student model এর সবগুলো object student variable এর মধ্যে রাখলাম।

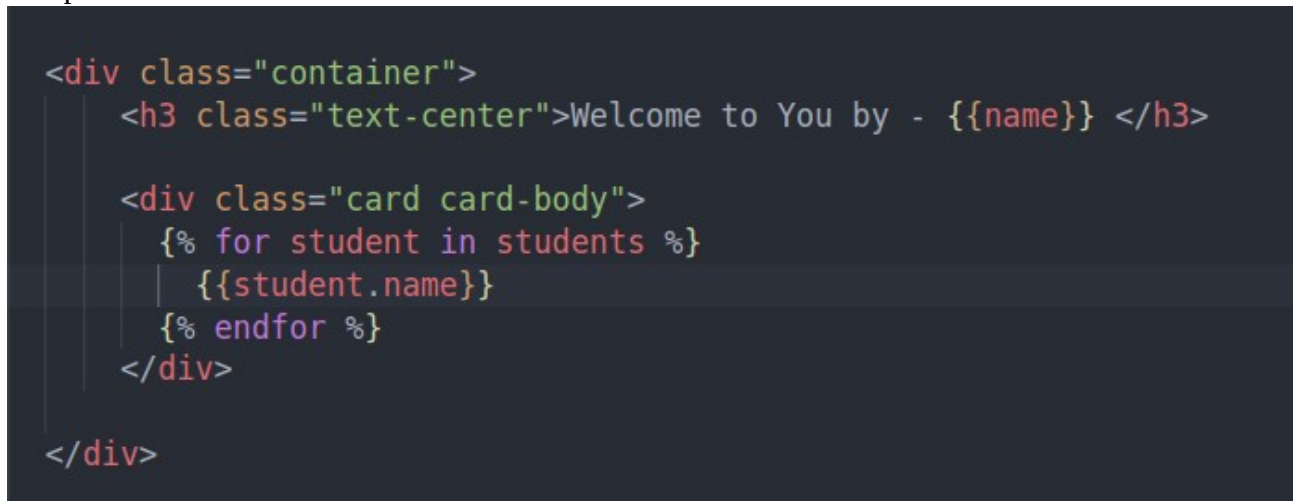
আগে যেভাবে আমার নামটি template এ পাঠাইছিলাম সেইভাবে Student object টি কেও পাঠালাম।

আমরা student variable টিকে print করলে এরকম দেখতে পারবো।

```
<QuerySet [<Student: Badsha>, <Student: Sazzad>, <Student: Musa>]>
[31/Jan/2023 08:10:07] "GET / HTTP/1.1" 200 1304
```

এখানে student variable এর মধ্যে অনেকগুলো student object আছে।

## Template



The screenshot shows a Django template code snippet with the following HTML and Django template language (DTL) code:

```
<div class="container">
  <h3 class="text-center">Welcome to You by - {{name}} </h3>

  <div class="card card-body">
    {% for student in students %}
      | {{student.name}}
    {% endfor %}
  </div>
</div>
```

যদি আমাদের অনেকগুলো ভালুএ থাকে তাহলে আমরা সেগুলো লুপ চালিয়ে দাখাবো।

এখানে আমাদের student object এর মধ্যে অনেকগুলো student এর object ছিল। তাই আমরা লুপ দিয়ে প্রত্যেক student এর value গুলো শো করবো।

template এর মধ্যে লেখা গুলো শো করার এই নিয়মকে বলে jinja template tag.

Template এর মধ্যে লুপ এর syntax হল।

```
{% for I in variable %}
```

```
{% endfor %}
```

আর কোনও variable এর value দেখাতে ছাইলে `{{ variable }}` এভাবে দেখাবো।

```
15
16
17     <div class="container">
18         <h3 class="text-center">Welcome to You by - {{name}} </h3>
19
20         {% for student in students %}
21         <div class="card card-body my-3">
22             {{student.name}}
23             <p>{{student.student_id}}</p>
24             <p>{{student.phone}}</p>
25             <p>{{student.address}}</p>
26         </div>
27         {% endfor %}
28
29     </div>
30
31
```

এখন যদি আমরা আমাদের ওয়েবপেজ এ যাই তাহলে আমরা নিচের মতো দেখতে পাবো।

Welcome to You by - badsha

Badsha  
12011077

Barishal

Sazzad  
12012121

Barishal

Musa  
2153153

Potuakhali