

First of all, create a student model inside (models.py)

Create your models here.

class Student(models.Model):

id = models.AutoField(primary_key=True)

name = models.CharField(max_length=200)

email= models.EmailField(max_length=200)

class Meta:

db_table = "students"

```
1  from django.db import models
2
3  # Create your models here.
4
5  # Create your models here.
6  class Student(models.Model):
7      id = models.AutoField(primary_key=True)
8      name = models.CharField(max_length=200)
9      email= models.EmailField(max_length=200)
10
11      class Meta:
12          db_table = "students"
13
```

Now go the the admin.py file and register your created model here

from django.contrib import admin

from .models import Student

Register your models here.

admin.site.register(Student)

```
1 from django.contrib import admin
2
3 from .models import Student
4 # Register your models here.
5 admin.site.register(Student)
6 |
```

Now run this code

\$ python manage.py makemigrations

```
C:\Users\Mr Hacker\Desktop\insert>python manage.py makemigrations
Migrations for 'create':
  create\migrations\0001_initial.py
    - Create model Student
```

\$ python manage.py migrate

```
C:\Users\Mr Hacker\Desktop\insert>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, create, sessions
Running migrations:
  Applying create.0001_initial... OK
```

```
C:\Users\Mr Hacker\Desktop\create\crud>python manage.py makemigrations
Migrations for 'crudapp':
  crudapp\migrations\0001_initial.py
    - Create model Student

C:\Users\Mr Hacker\Desktop\create\crud>
```

Creating an admin user

First, we'll need to create a user who can login to the admin site. Run the following command:

\$ python manage.py createsuperuser

```
C:\Users\Mr Hacker\Desktop\create\crud>python manage.py createsuperuser
Username (leave blank to use 'mrhacker'): bita639
Email address: 01625326156dj@gmail.com
Password:
Password (again):
Superuser created successfully.

C:\Users\Mr Hacker\Desktop\create\crud>
```

Start the development server

The Django admin site is activated by default. Let's start the development server and explore it.

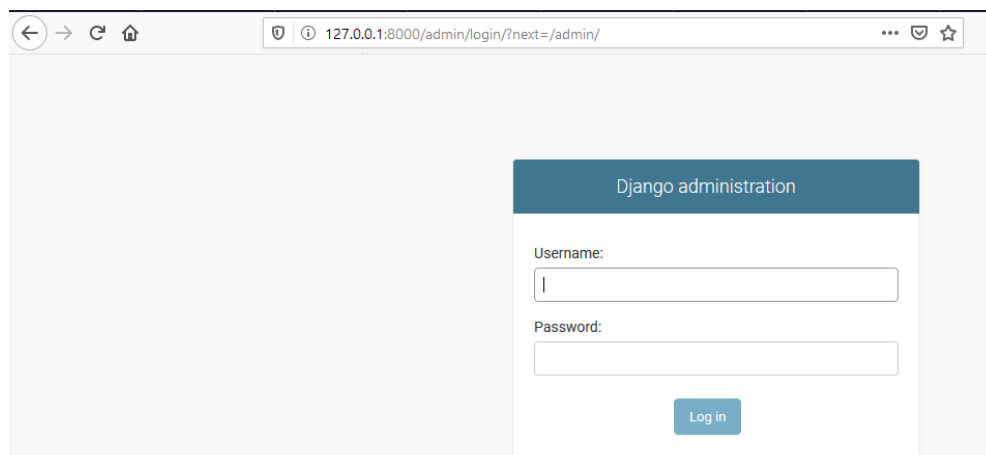
Recall from Tutorial 1 that you start the development server like so:

```
$ python manage.py runserver
```

Now, open a Web browser and go to “/admin/” on your local domain – e.g., <http://127.0.0.1:8000/admin/>. You should see the admin's login screen:

```
C:\Users\Mr Hacker\Desktop\create\crud>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 27, 2019 - 23:17:37
Django version 2.2.5, using settings 'crud.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/admin/login/?next=/admin/`. The page content is the Django administration login interface. It features a dark blue header with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". A blue "Log in" button is positioned below the password field. The background of the page is a light gray.

Django administration

Username:

bita639

Password:

●●●●●●●●●●

Log in

127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Change

Users

+ Add

Change

CRUDAPP

Students

+ Add

Change

Recent actions

My actions

+ Student object (2)

Student

+ Student object (1)

Student

Now click on student and insert data

127.0.0.1:8000/admin/crudapp/student/2/change/

Django administration

WELCOME, BITA639. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Crudapp > Students > Student object (2)

Change student

HISTORY

Name:

Nazmul Hossain

Email:

admin@letsgotravelsbd.com

Delete

Save and add another

Save and continue editing

SAVE

now go to index.html file and crate the view for fetching data.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>{{title}}</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.
min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"><
/script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.mi
n.js"></script>
</head>
<body>

<div class="container">
  <h2>Bordered Table</h2>
  <p>The .table-bordered class adds borders to a table:</p>
  <table class="table table-bordered">
    <thead>
      <tr>
```

```
<th>Id</th>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
  {%for std in students %}
  <tr>
    <td>{{std.id}}</td>
    <td>{{std.name}}</td>
    <td>{{std.email}}</td>
  </tr>
  {% endfor %}
</tbody>
</table>
</div>

</body>
</html>
```

This is the code inside the index.html file

Now save the file and run the server to check

\$ manage.py runserver



Bordered Table

The `.table-bordered` class adds borders to a table:

Id	Name	Email
2	Nazmul Hossain	admin@letsgotravelsbd.com
1	saiful Islam	01625326156dj@gmail.com

BOOM !!!!!