

Creating a Model to Add Database Service & Django Administration App

Step-1: Create Project

```
(base) C:\djproject>django-admin startproject project
```

Step-2: Create App (After get into project folder)

```
(base) C:\djproject\project>python manage.py startapp app
```

Step-3: Edit project Settings.py file to add your app name (Here app name is app)

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'app'  
]
```

Step 4: Add models in app Models.py file

```

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      emailid=models.EmailField(max_length=50)
7      contact_no=models.CharField(max_length=11)
8      DOB=models.DateTimeField()
9
10     def __str__(self):
11         return self.name
12

```

Step 5: Do makemigrations and migrate with manage.py

```

(base) C:\djproject\project>python manage.py makemigrations
Migrations for 'app':
  app\migrations\0001_initial.py
    - Create model student

```

```

(base) C:\djproject\project>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, app, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK

```

Step 7: Add model information in app Admin.py file

```

from django.contrib import admin

# Register your models here.

from .models import *

admin.site.register(student)

```

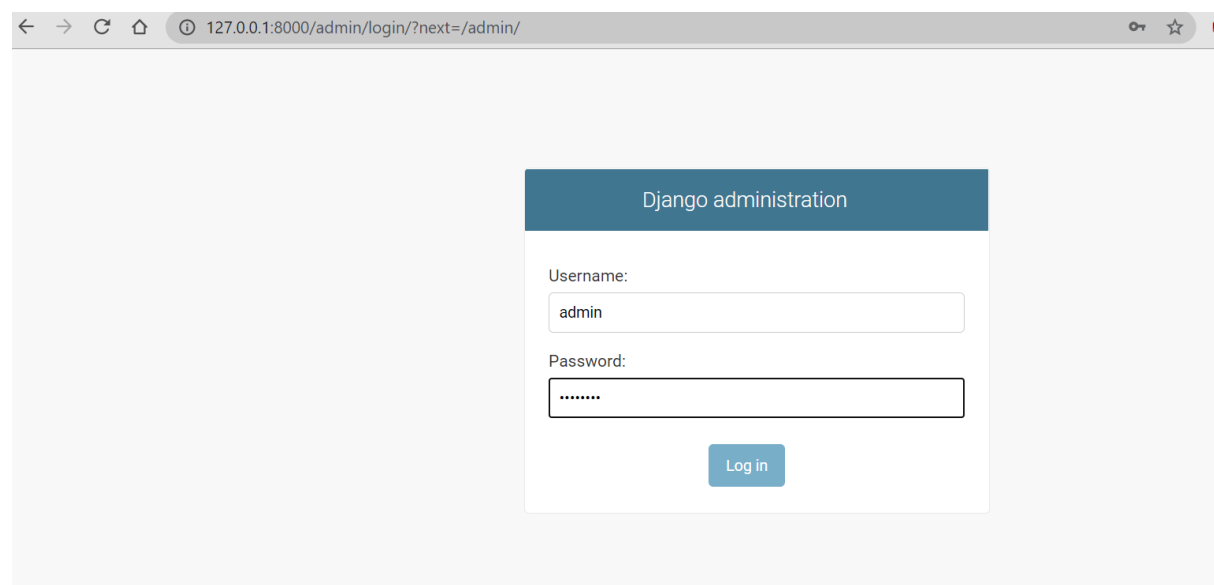
Step 6: Before starting your server – Create Superuser for your project

```
(base) C:\djproject\project>python manage.py createsuperuser
Username (leave blank to use 'prabu'): admin
Email address: admin@example.com
Password:
Password (again):
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Step 7: Start server

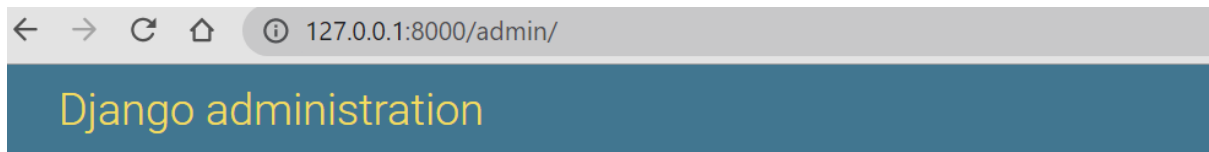
```
(base) C:\djproject\project>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

Step 8: Login into Admin Application from browser



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 blue header bar with the text "Django administration". Below this, there is a white box containing the login form. The form has two input fields: "Username:" with the value "admin" and "Password:" with masked characters ".....". A blue "Log in" button is positioned below the password field.

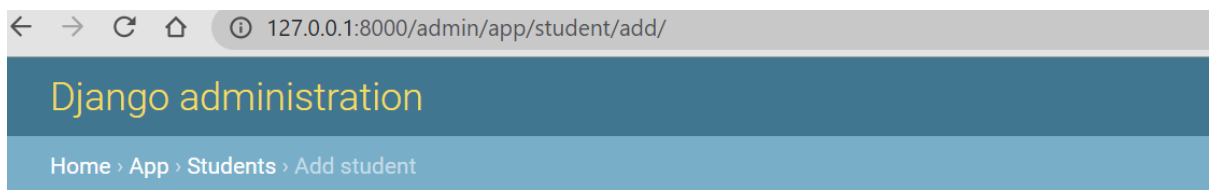
Step 9: Admin Page



Site administration

APP		
Students	+ Add	✎ Change
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

Step 10: You can add student information (select students from admin page)



Add student

Name:	<input type="text"/>						
Emailid:	<input type="text"/>						
Contact no:	<input type="text"/>						
DOB:	<table><tr><td>Date:</td><td><input type="text"/></td><td>Today </td></tr><tr><td>Time:</td><td><input type="text"/></td><td>Now </td></tr></table>	Date:	<input type="text"/>	Today	Time:	<input type="text"/>	Now
Date:	<input type="text"/>	Today					
Time:	<input type="text"/>	Now					
Note: You are 5.5 hours ahead of server time.							
<div></div>							