Software_Engineering

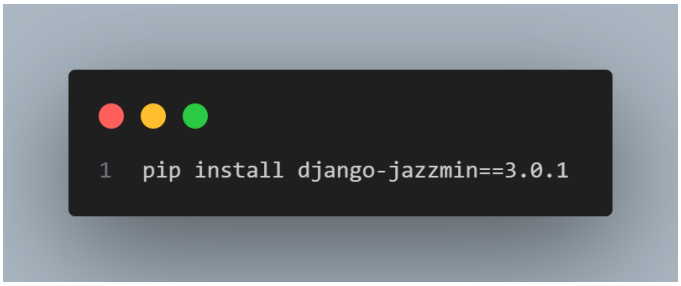
Lab7

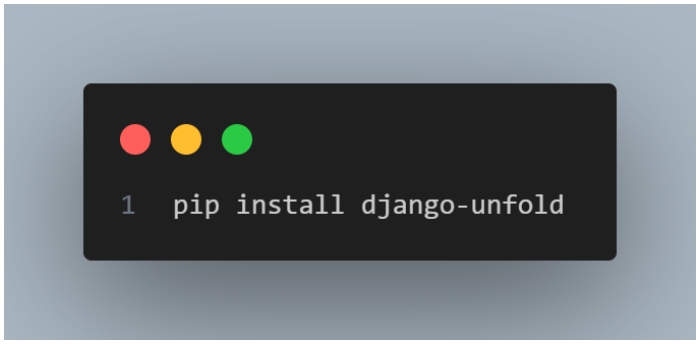
Heyam Hameed

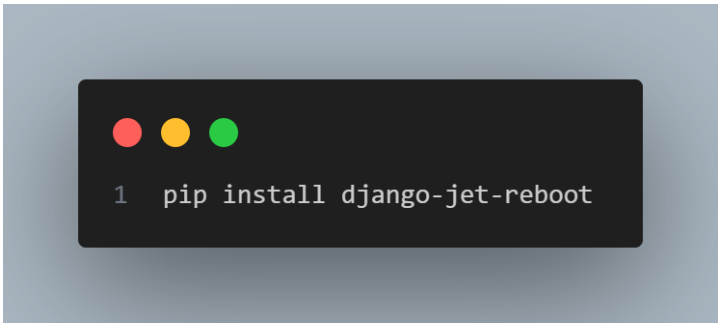
مقارنة بين أنواع الثيمات :

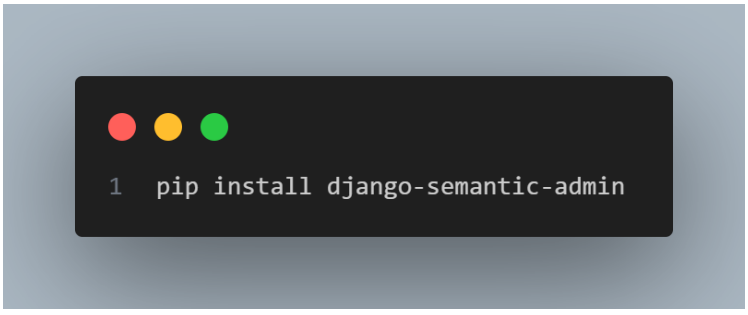
الثيم	شكل الواجهة	المميزات البارزة
Jazzmin	تصميم حديث وأنيق يشبه لوحة تحكم التطبيقات، ألوان قابلة للتخصيص، قوائم جانبية قابلة للطي	يدعم Dark Mode، تخصيص القوائم واللوجو، سهل التطبيق على أي مشروع
Unfold	تصميم مرن ونظيف، قائم على Bootstrap، قوائم جانبية بسيطة ومرتبة، أيقونات واضحة.	سريع التحميل، يدعم RTL، مناسب للمشاريع التعليمية
Jet	تصميم احترافي وجذاب، يضيف Dashboardمع إحصائيات، أيقونات متحركة، bookmarksلكل موديل.	Dashboardديناميكي، إدارة Bookmarks وPinned Apps، أفضل لوحات الإدارة الكبيرة.
Semantic Admin	تصميم Minimalومرتب، يعتمد على Semantic UI، قوائم واضحة، ألوان هادئة	خفيف، لا يحتاج جداول خاصة للثيم، سهل التركيب ويعمل مع أي Admin.

أولا نثبت المكاتب اللازمة لكل ثيم :









ثانيا تفعيل التطبيق:

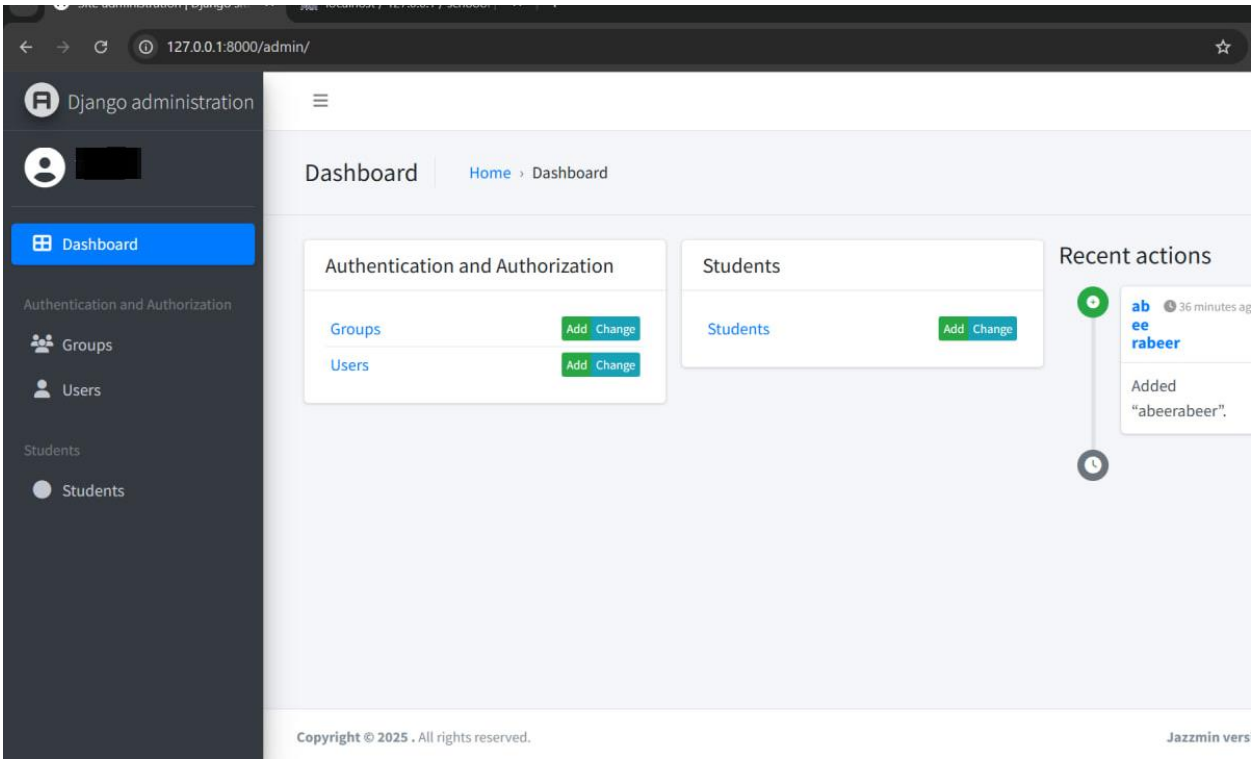
نفعل التطبيقات داخل ملفات ال sitting.py التي في المشروع

ثالثا تشغيل السيرفر

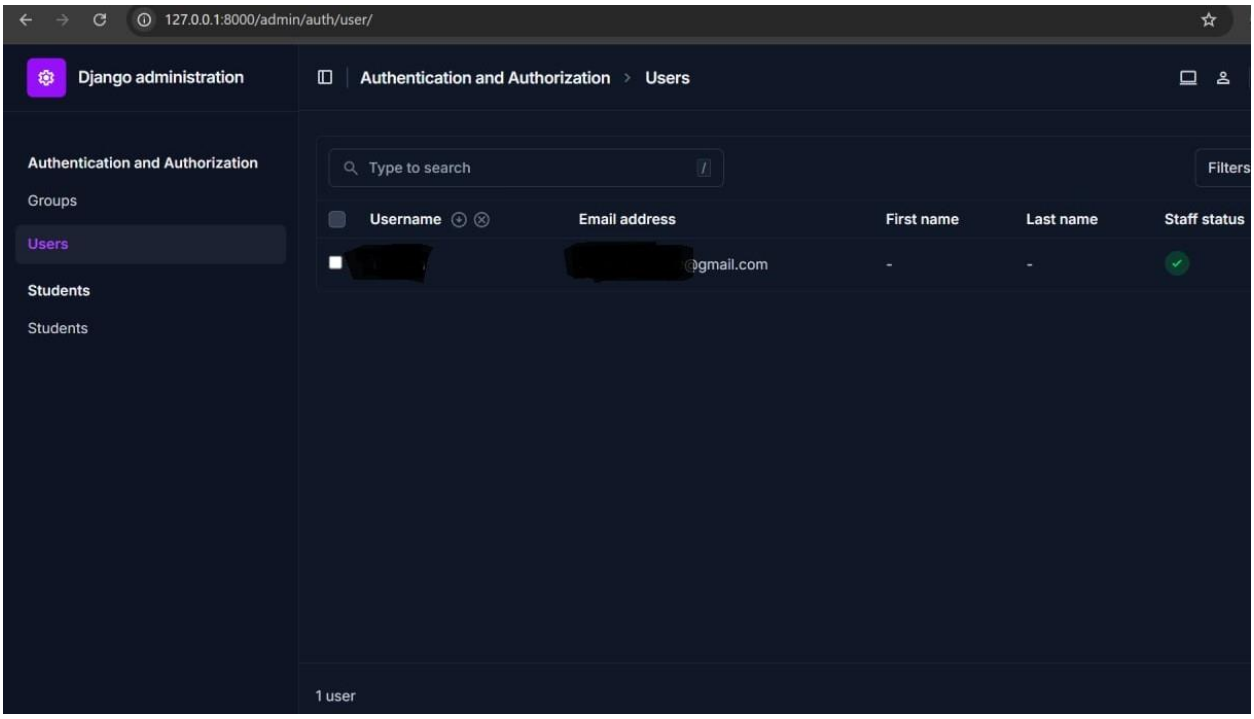
ملاحظة:

هناك ثيمات تحتاج جمع الملفات الثابتة (Static Files)

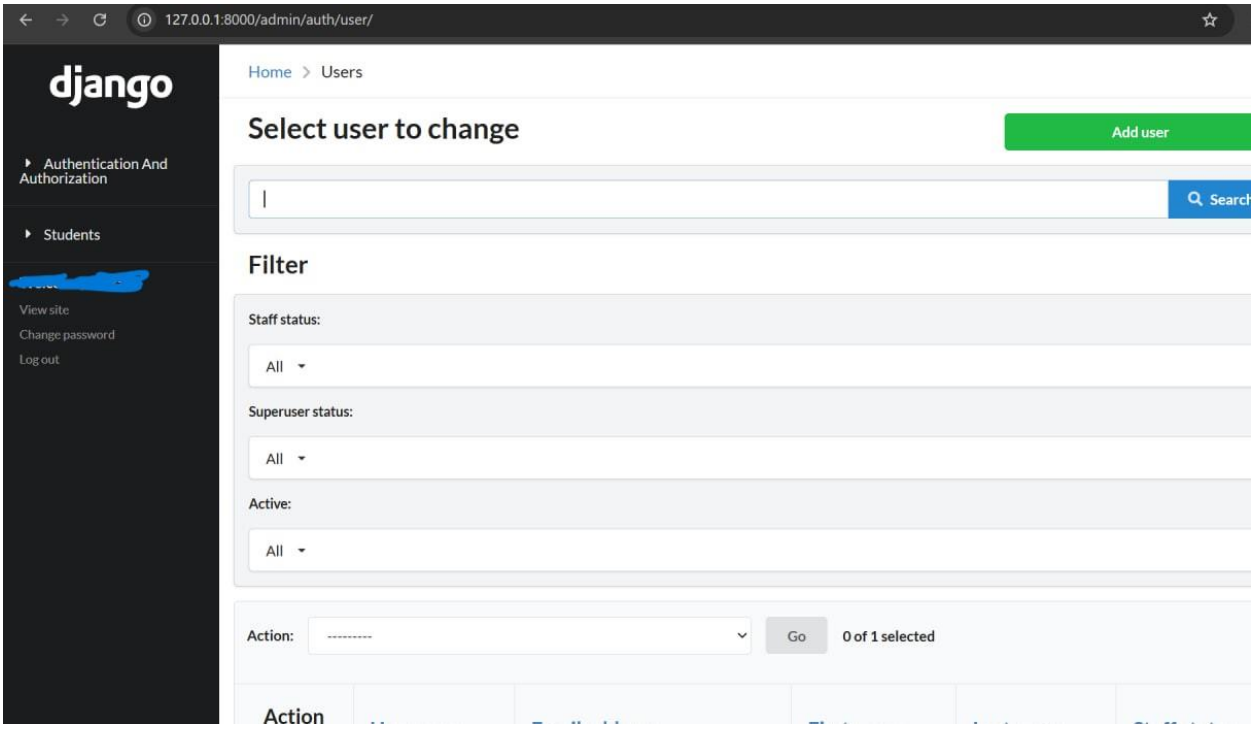
١-Jazzmin



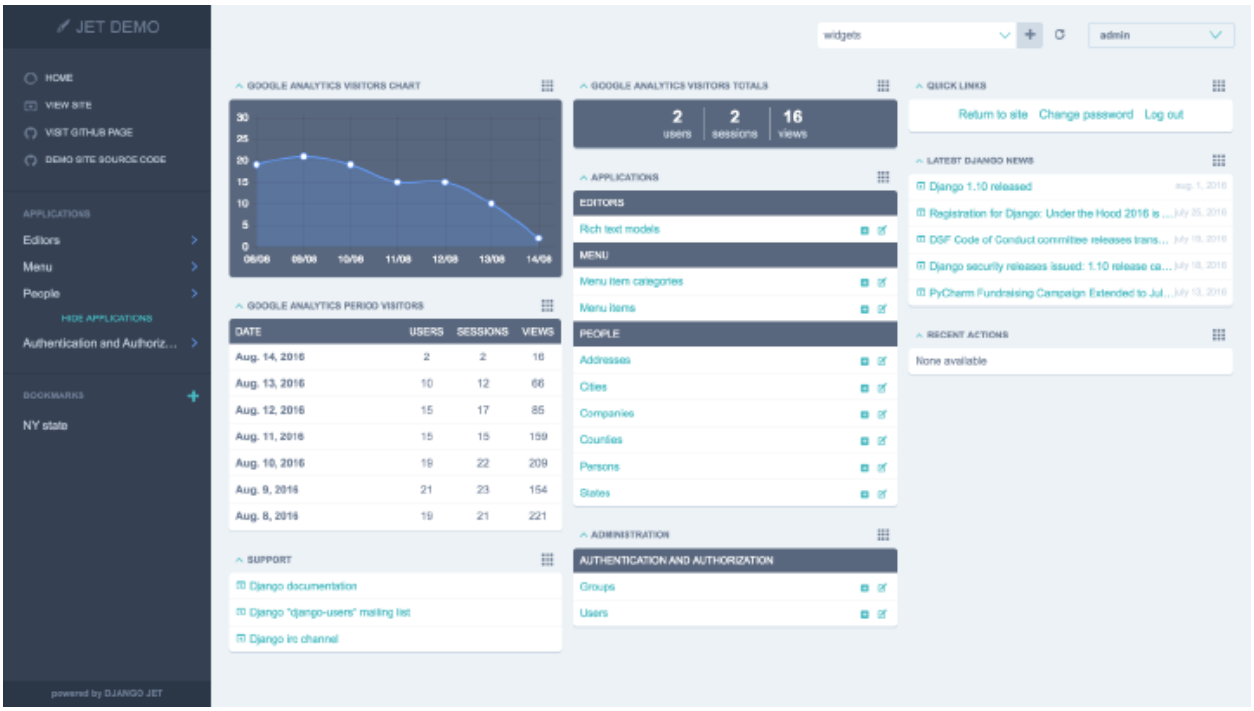
Unfold-۲



Semantic Admin-۳



Jet-٤


JET DEMO
HOME
VIEW SITE
VISIT GITHUB PAGE
DEMO SITE SOURCE CODE
APPLICATIONS
Editors
Menu
People
HIDE APPLICATIONS
Authentication and Authoriz...
BOOKMARKS
NY state
powered by DJANGO JET
widgets
admin
GOOGLE ANALYTICS VISITORS CHART
GOOGLE ANALYTICS VISITORS TOTALS
2 users
2 sessions
16 views
QUICK LINKS
Return to site
Change password
Log out
APPLICATIONS
EDITORS
Rich text models
MENU
Menu item categories
Menu items
PEOPLE
Addresses
Cities
Companies
Counties
Persons
States
LATEST DJANGO NEWS
Django 1.10 released
Registration for Django: Under the Hood 2016 is ...
DSF Code of Conduct committee releases trans...
Django security releases issued: 1.10 release ca...
PyCharm Fundraising Campaign Extended to Jul...
GOOGLE ANALYTICS PERIOD VISITORS
DATE USERS SESSIONS VIEWS
Aug. 14, 2016 2 2 16
Aug. 13, 2016 10 12 66
Aug. 12, 2016 15 17 85
Aug. 11, 2016 15 15 159
Aug. 10, 2016 19 22 209
Aug. 9, 2016 21 23 154
Aug. 8, 2016 19 21 221
ADMINISTRATION
AUTHENTICATION AND AUTHORIZATION
Groups
Users
RECENT ACTIONS
None available
SUPPORT
Django documentation
Django "django-users" mailing list
Django irc channel

إضافة علاقات لقاعدة البيانات:

١-One to One: علاقة **طالب – ملف صحي:**

في تطبيق الطلاب نكتب الكود التالي :

```python
from django.db import models

class Student(models.Model):
    name = models.CharField(max_length=100)
    age = models.IntegerField()
    grade = models.CharField(max_length=50)

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

في تطبيق الملفات الصحية:

```python
from django.db import models
from students.models import Student   # نربط بمودل الطالب من التطبيق الآخر

class HealthRecord(models.Model):
    student = models.OneToOneField(Student, on_delete=models.CASCADE)
    blood_type = models.CharField(max_length=3)
    allergies = models.TextField(blank=True, null=True)
    chronic_diseases = models.TextField(blank=True, null=True)

    def __str__(self):
        return f"Health Record of {self.student.name}"
```

٢-Many-to-Many: علاقة **طالب ــ مادة**:

في تطبيق المواد:

```python
from django.db import models

class Subject(models.Model):
    name = models.CharField(max_length=100)
    code = models.CharField(max_length=20, unique=True)

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

في تطبيق الطلاب:

```python
from django.db import models
from subjects.models import Subject   # استدعاء المودل من تطبيق المواد

class Student(models.Model):
    name = models.CharField(max_length=100)
    age = models.IntegerField()
    grade = models.CharField(max_length=50)

    subjects = models.ManyToManyField(Subject, related_name="students")

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

٣-One-to-Many: العلاقة **معلم – طلاب**:

في تطبيق المعلمين:

```python
from django.db import models

class Teacher(models.Model):
    name = models.CharField(max_length=100)
    subject = models.CharField(max_length=100)

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

في تطبيق الطلاب:

```python
from django.db import models
from teachers.models import Teacher   # استدعاء مودل المعلم

class Student(models.Model):
    name = models.CharField(max_length=100)
    age = models.IntegerField()
    grade = models.CharField(max_length=50)

    teacher = models.ForeignKey(
        Teacher,
        on_delete=models.CASCADE,
        related_name="students"
    )

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

للتنفيذ:

١- تشغيل أوامر إنشاء الجداول
٢- تسجليهم في لوحة الإدارة
٣- من **Django Admin** نربط كل طالب بمعلم من القائمة المنسدلة و هكذا.