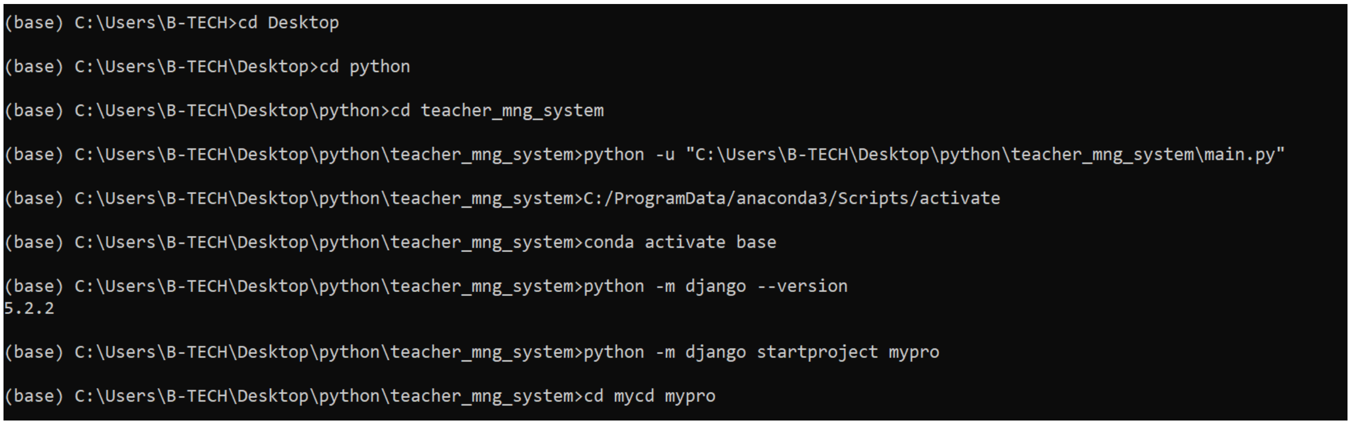
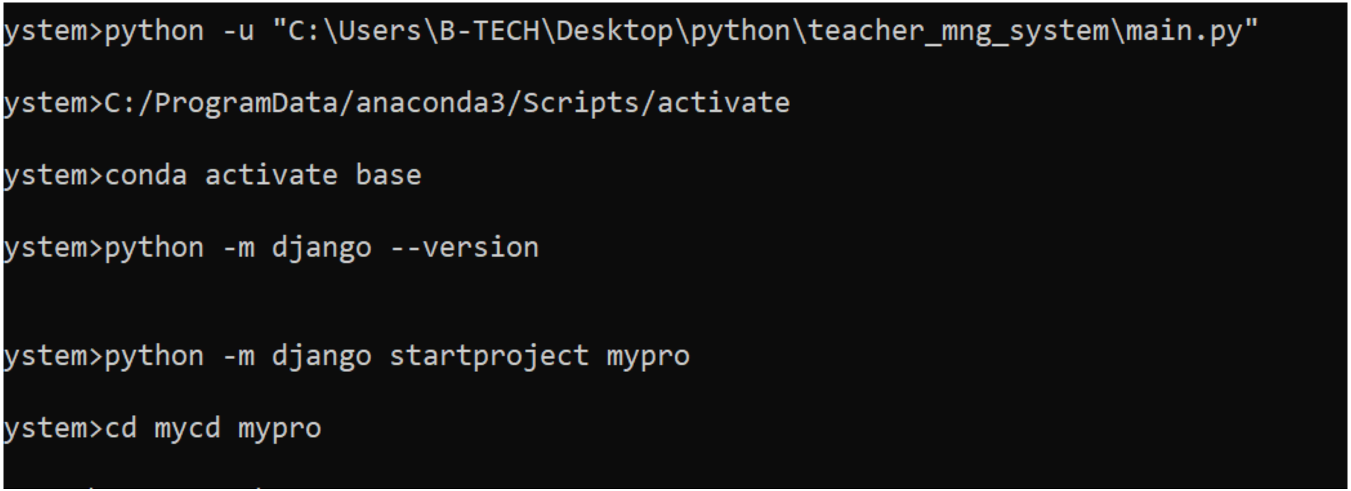
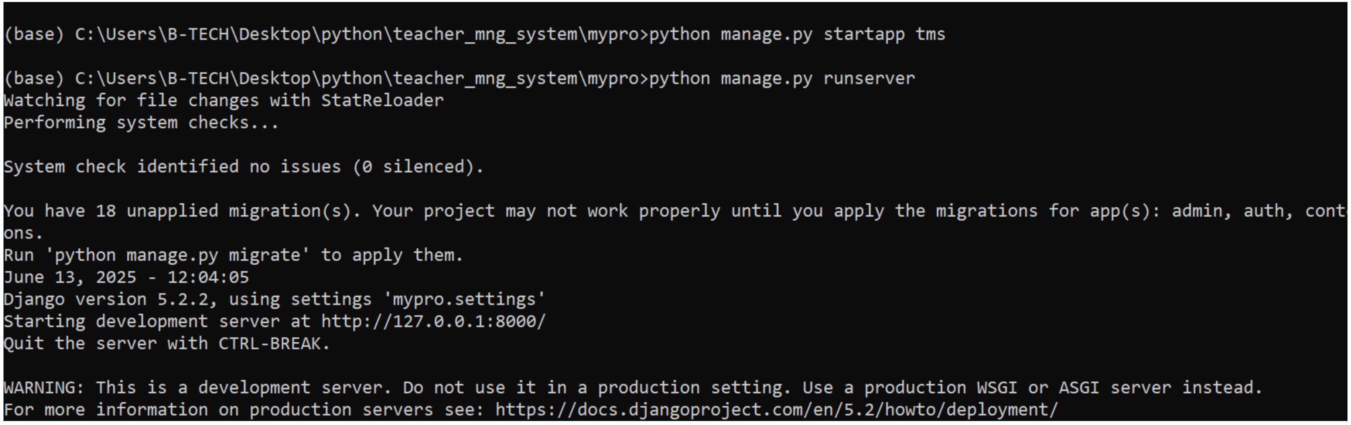


```
(base) C:\Users\B-TECH>cd Desktop

(base) C:\Users\B-TECH\Desktop>cd python

(base) C:\Users\B-TECH\Desktop\python>cd teacher_mng_system

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system>python -u "C:\Users\B-TECH\Desktop\python\teacher_mng_system\main.py"

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system>C:/ProgramData/anaconda3/Scripts/activate

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system>conda activate base

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system>python -m django --version
5.2.2

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system>python -m django startproject mypro

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system>cd mycd mypro
```

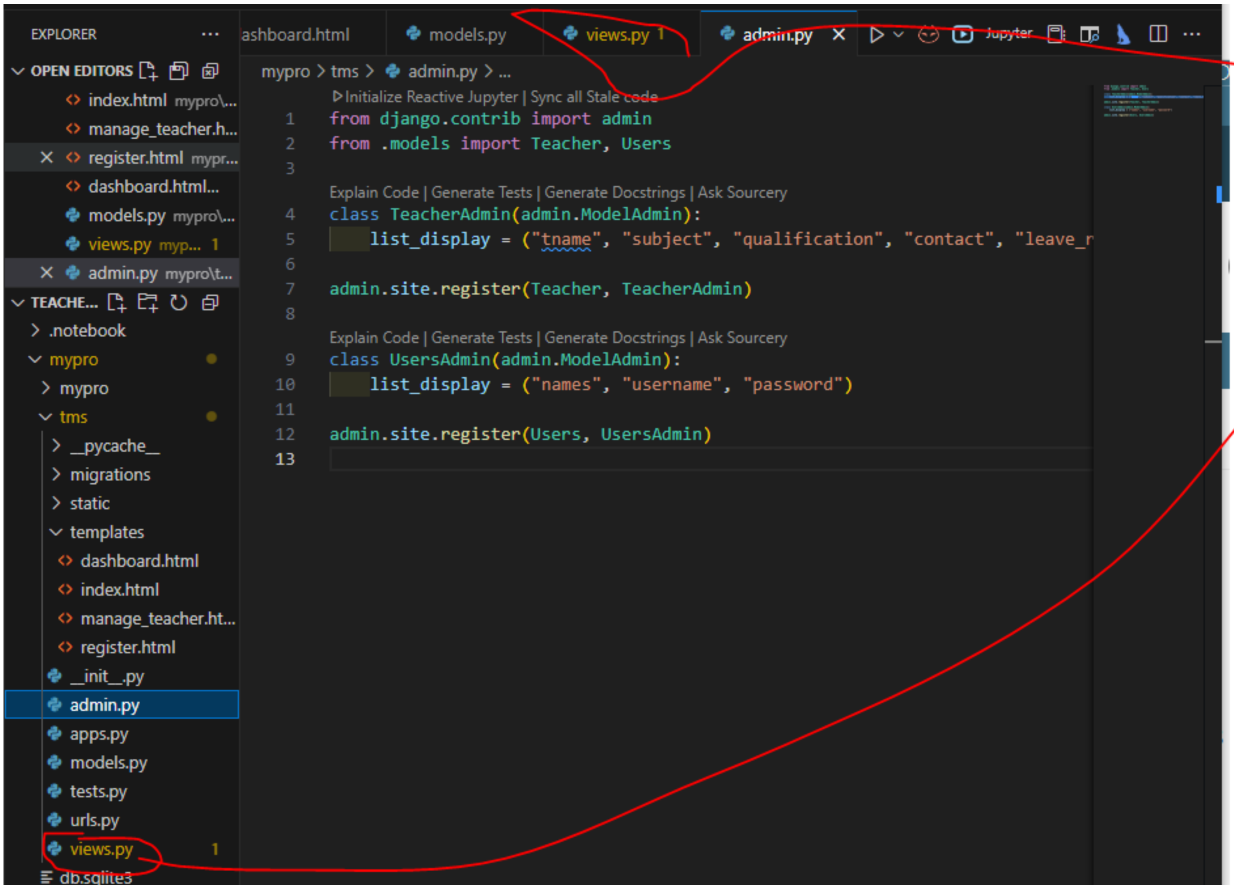


```
ystem>python -u "C:\Users\B-TECH\Desktop\python\teacher_mng_system\main.py"

ystem>C:/ProgramData/anaconda3/Scripts/activate

ystem>conda activate base

ystem>python -m django --version


ystem>python -m django startproject mypro

ystem>cd mycd mypro
```

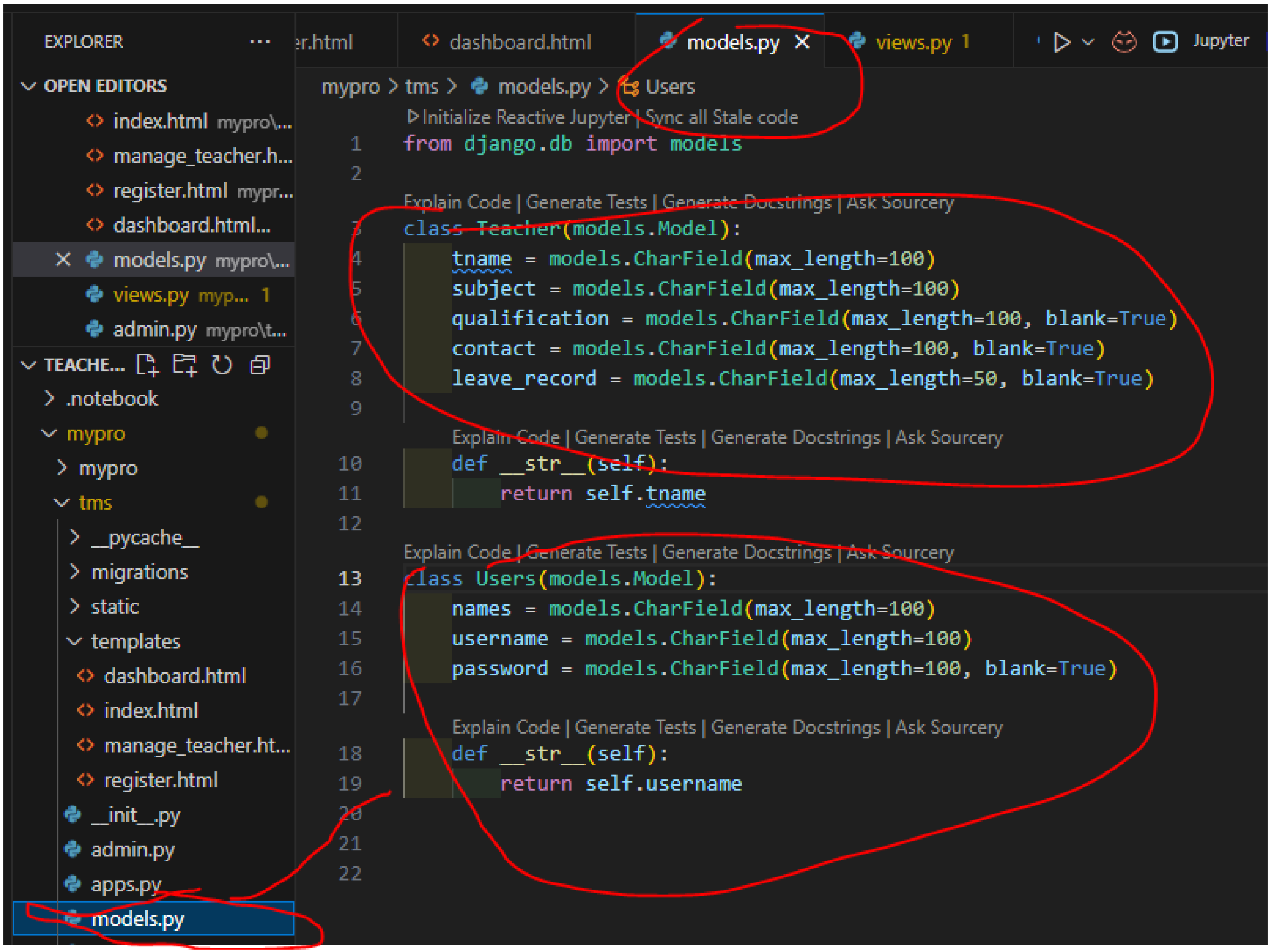


```
(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py startapp tms

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, cont
ons.
Run 'python manage.py migrate' to apply them.
June 13, 2025 - 12:04:05
Django version 5.2.2, using settings 'mypro.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
```

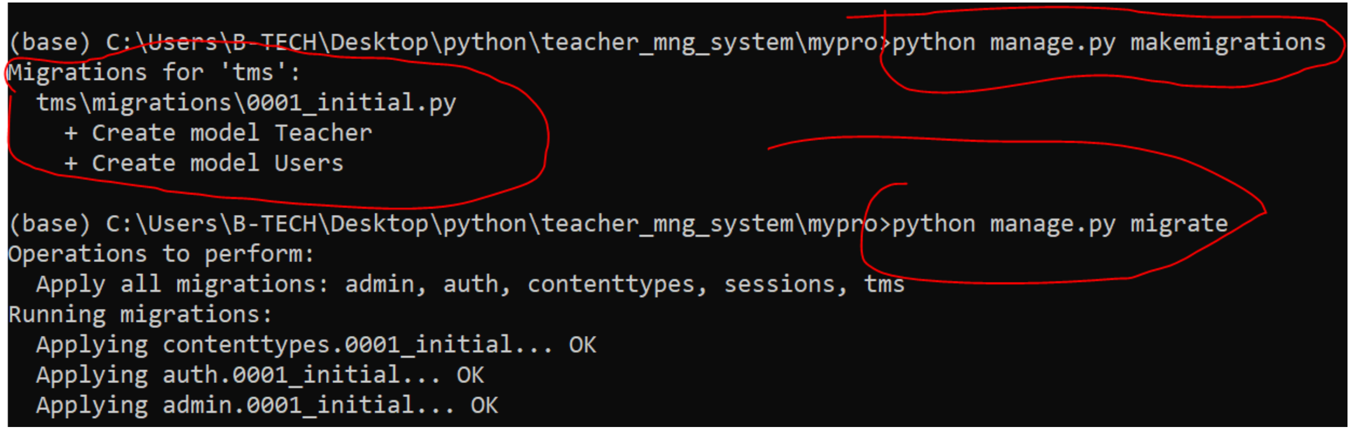
Database



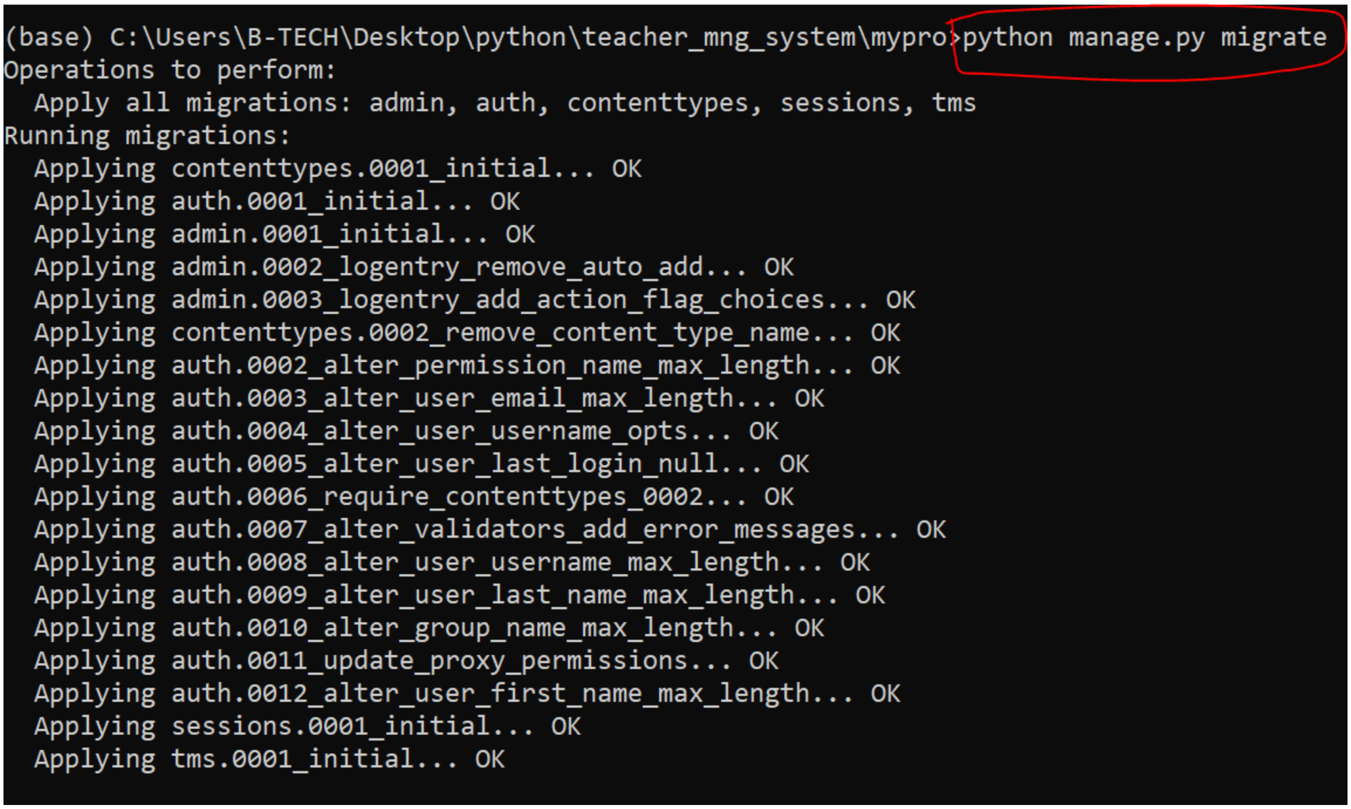
```python
from django.contrib import admin
from .models import Teacher, Users

class TeacherAdmin(admin.ModelAdmin):
    list_display = ("tname", "subject", "qualification", "contact", "leave_r

admin.site.register(Teacher, TeacherAdmin)

class UsersAdmin(admin.ModelAdmin):
    list_display = ("names", "username", "password")

admin.site.register(Users, UsersAdmin)

```



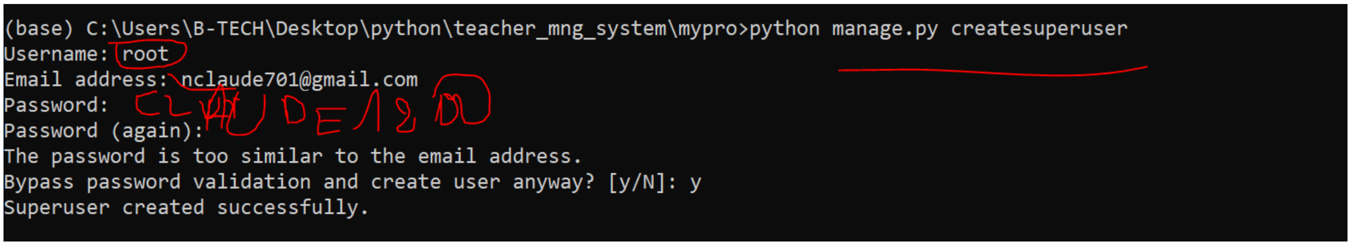
```python
from django.db import models

class Teacher(models.Model):
    tname = models.CharField(max_length=100)
    subject = models.CharField(max_length=100)
    qualification = models.CharField(max_length=100, blank=True)
    contact = models.CharField(max_length=100, blank=True)
    leave_record = models.CharField(max_length=50, blank=True)

    def __str__(self):
        return self.tname

class Users(models.Model):
    names = models.CharField(max_length=100)
    username = models.CharField(max_length=100)
    password = models.CharField(max_length=100, blank=True)

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



```
(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py makemigrations
Migrations for 'tms':
  tms\migrations\0001_initial.py
    + Create model Teacher
    + Create model Users

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, tms
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
```

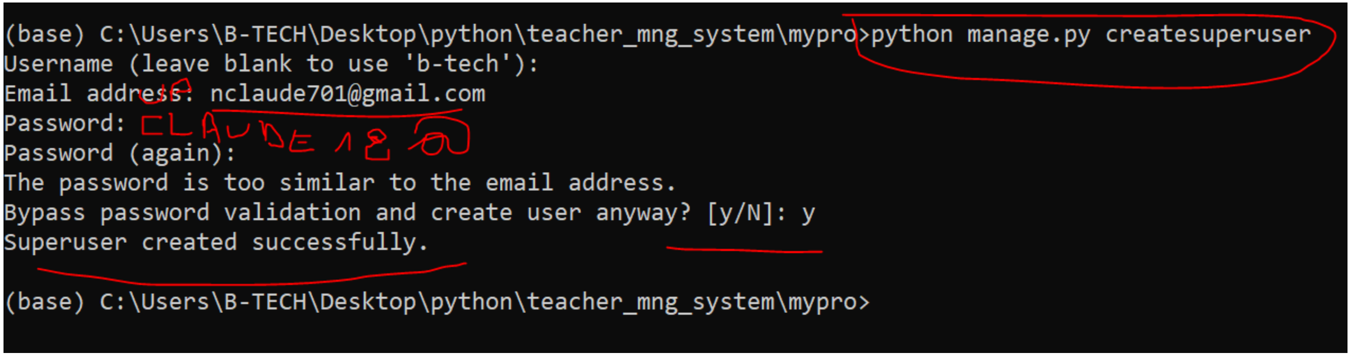


```
(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, tms
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
  Applying tms.0001_initial... OK
```



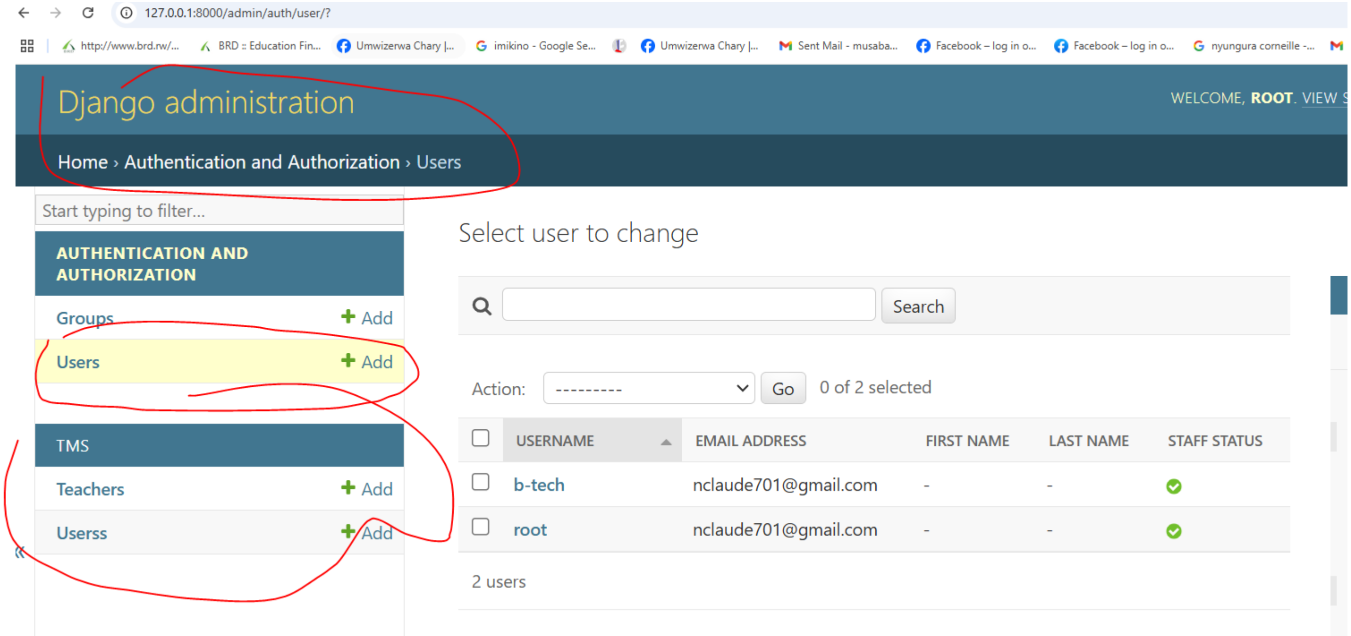
```
(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py createsuperuser
Username: root
Email address: nclaude701@gmail.com
Password:
Password (again):
The password is too similar to the email address.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

CLAUDE12@



```
(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>python manage.py createsuperuser
Username (leave blank to use 'b-tech'):
Email address: nclaude701@gmail.com
Password:
Password (again):
The password is too similar to the email address.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

(base) C:\Users\B-TECH\Desktop\python\teacher_mng_system\mypro>
```

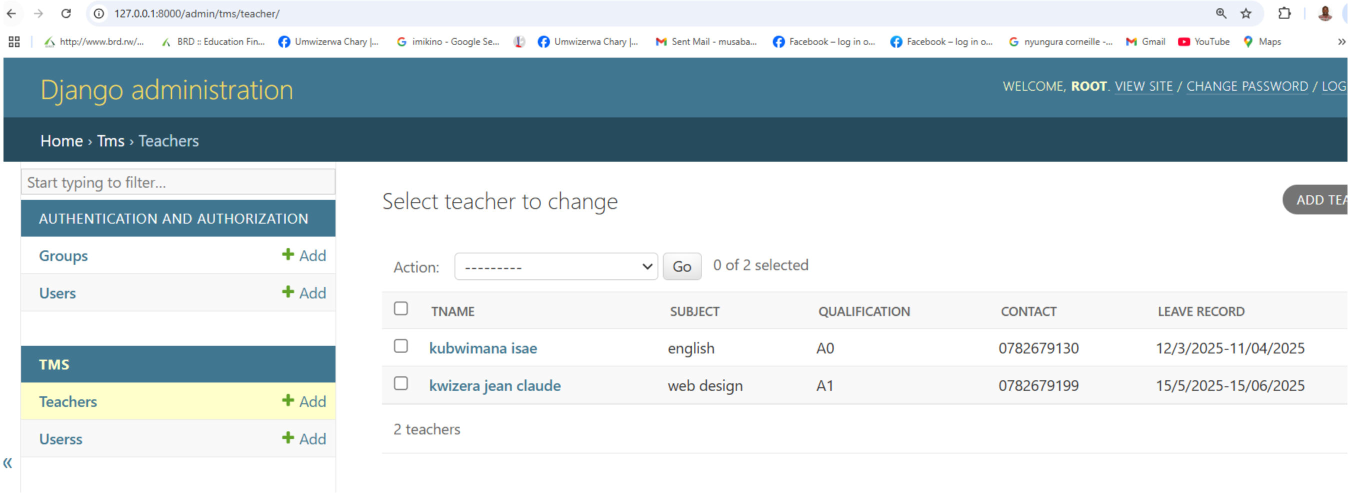
CLAUDE12@


127.0.0.1:8000/admin/auth/user/?
http://www.brd.rw/... BRD :: Education Fin... Umwizerwa Chary |... imikino - Google Se... Umwizerwa Chary |... Sent Mail - musaba... Facebook – log in o... Facebook – log in o... nyungura corneille -...
Django administration
WELCOME, ROOT. VIEW S
Home › Authentication and Authorization › Users
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
TMS
Teachers + Add
Userss + Add
«
Search
Action: --------- Go 0 of 2 selected

Select user to change

☐	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
☐	b-tech	nclaude701@gmail.com	-	-	✓
☐	root	nclaude701@gmail.com	-	-	✓

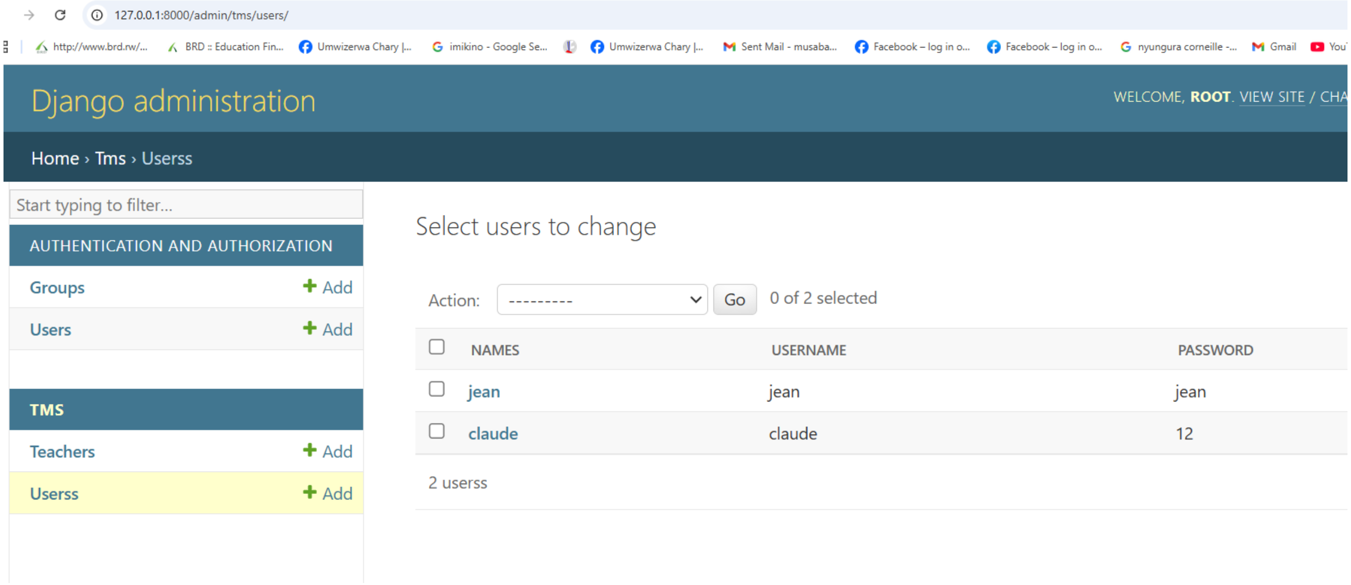
2 users


127.0.0.1:8000/admin/tms/teacher/
http://www.brd.rw/... BRD :: Education Fin... Umwizerwa Chary |... imikino - Google Se... Umwizerwa Chary |... Sent Mail - musaba... Facebook – log in o... Facebook – log in o... nyungura corneille -... Gmail YouTube Maps
Django administration
WELCOME, ROOT. VIEW SITE / CHANGE PASSWORD / LOG
Home › Tms › Teachers
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
TMS
Teachers + Add
Userss + Add
«
ADD TEA
Action: --------- Go 0 of 2 selected

Select teacher to change

☐	TNAME	SUBJECT	QUALIFICATION	CONTACT	LEAVE RECORD
☐	kubwimana isae	english	A0	0782679130	12/3/2025-11/04/2025
☐	kwizera jean claude	web design	A1	0782679199	15/5/2025-15/06/2025

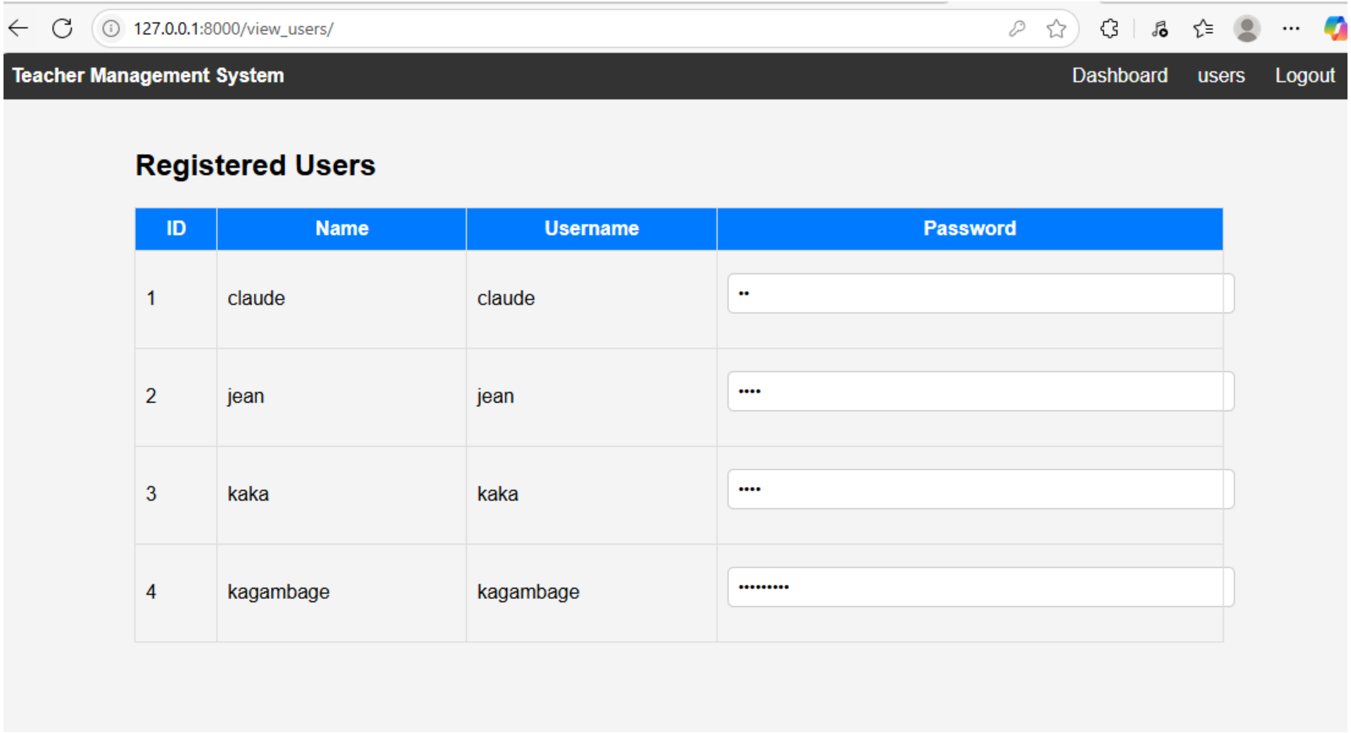
2 teachers


127.0.0.1:8000/admin/tms/users/
http://www.brd.rw/... BRD :: Education Fin... Umwizerwa Chary |... imikino - Google Se... Umwizerwa Chary |... Sent Mail - musaba... Facebook – log in o... Facebook – log in o... nyungura corneille -... Gmail You
Django administration
WELCOME, ROOT. VIEW SITE / CHA
Home › Tms › Userss
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
TMS
Teachers + Add
Userss + Add
Action: --------- Go 0 of 2 selected

Select users to change

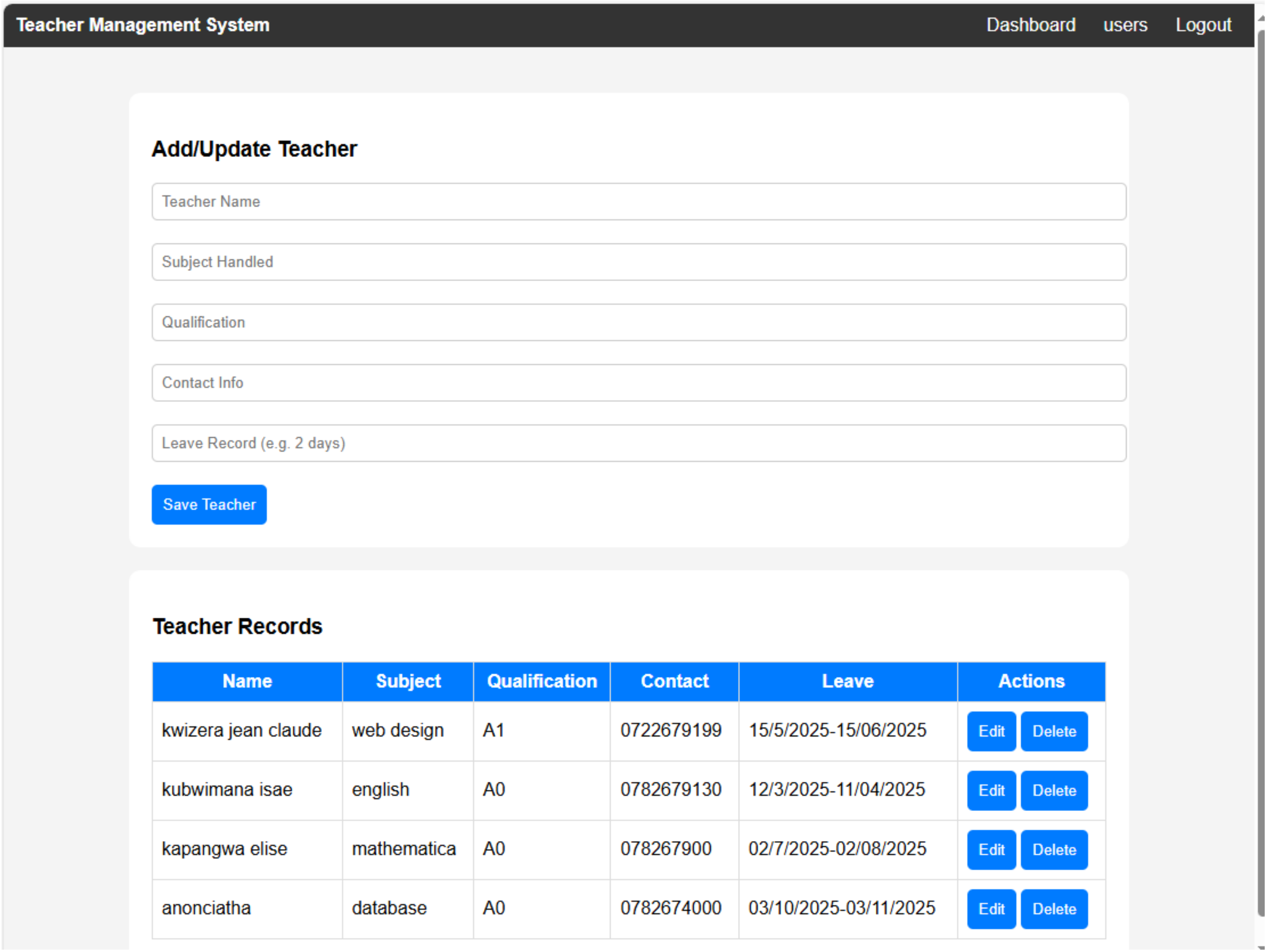
☐	NAMES	USERNAME	PASSWORD
☐	jean	jean	jean
☐	claude	claude	12

2 userss



Registered Users

ID	Name	Username	Password
1	claude	claude	••
2	jean	jean	••••
3	kaka	kaka	••••
4	kagambage	kagambage	••••••••••



Add/Update Teacher

Teacher Name

Subject Handled

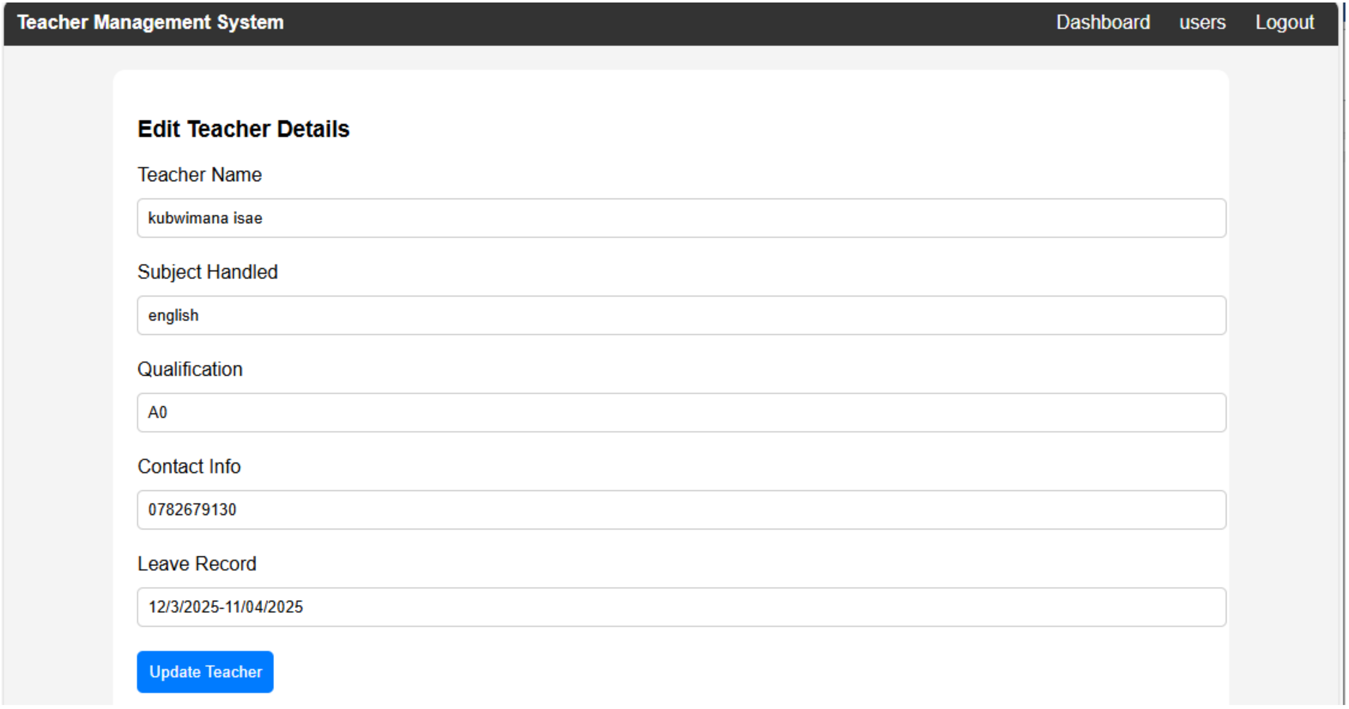
Qualification

Contact Info

Leave Record (e.g. 2 days)

Teacher Records

Name	Subject	Qualification	Contact	Leave	Actions
kwizera jean claude	web design	A1	0722679199	15/5/2025-15/06/2025	Edit Delete
kubwimana isae	english	A0	0782679130	12/3/2025-11/04/2025	Edit Delete
kapangwa elise	mathematica	A0	078267900	02/7/2025-02/08/2025	Edit Delete
anonciatha	database	A0	0782674000	03/10/2025-03/11/2025	Edit Delete



Edit Teacher Details

Teacher Name

kubwimana isae

Subject Handled

english

Qualification

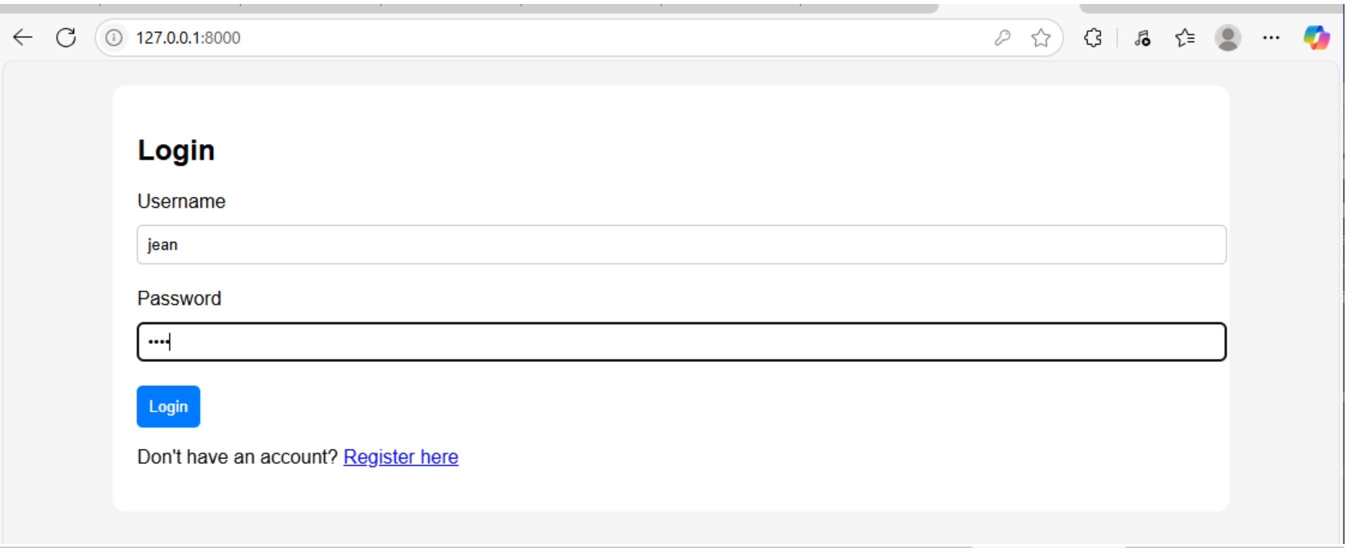
A0

Contact Info

0782679130

Leave Record

12/3/2025-11/04/2025



127.0.0.1:8000

Login

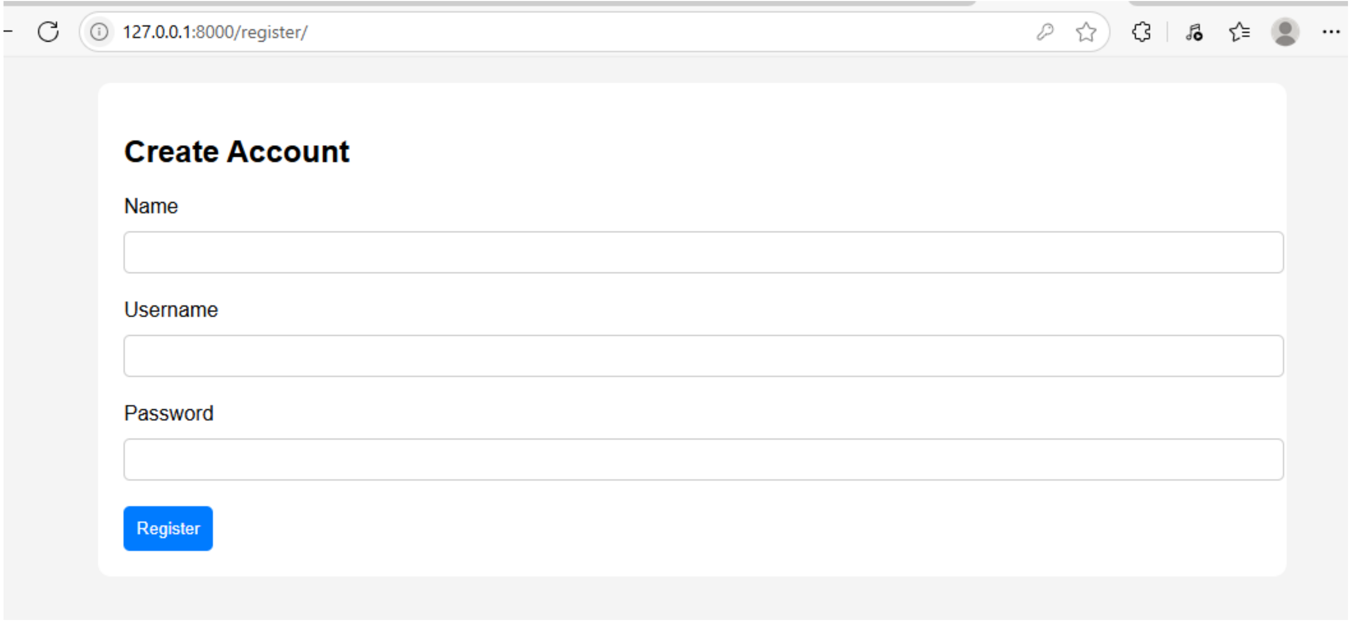
Username

jean

Password

••••

Don't have an account? Register here



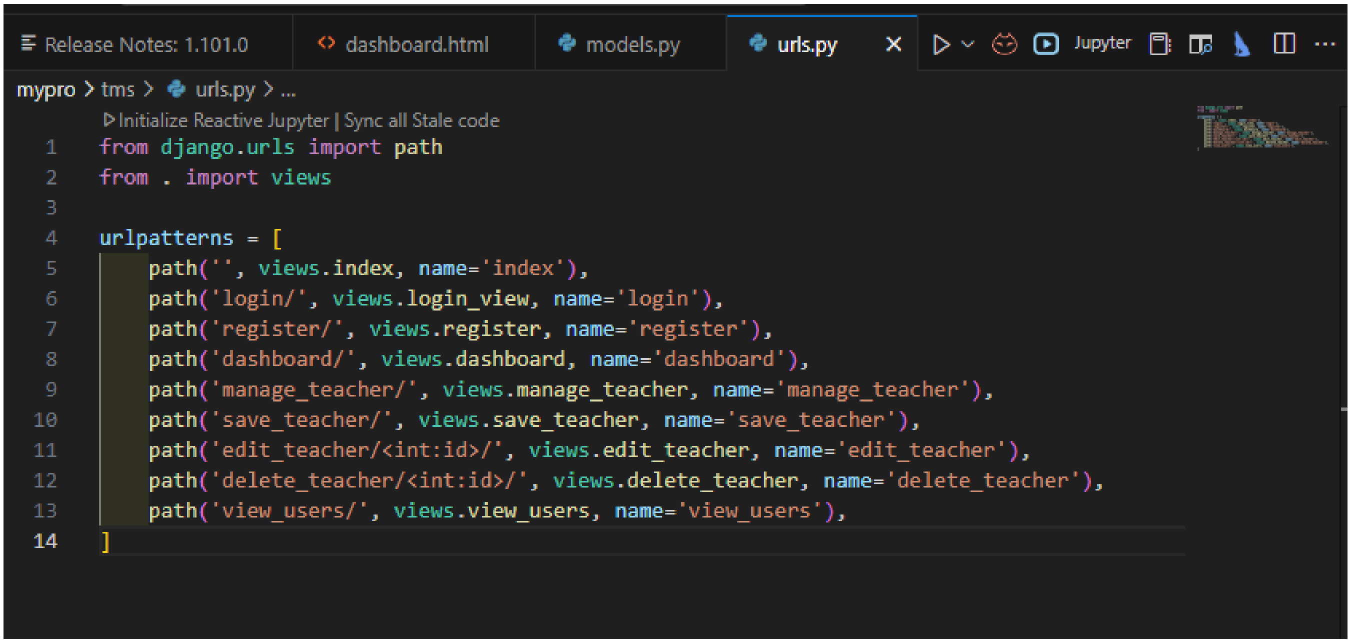
127.0.0.1:8000/register/

Create Account

Name

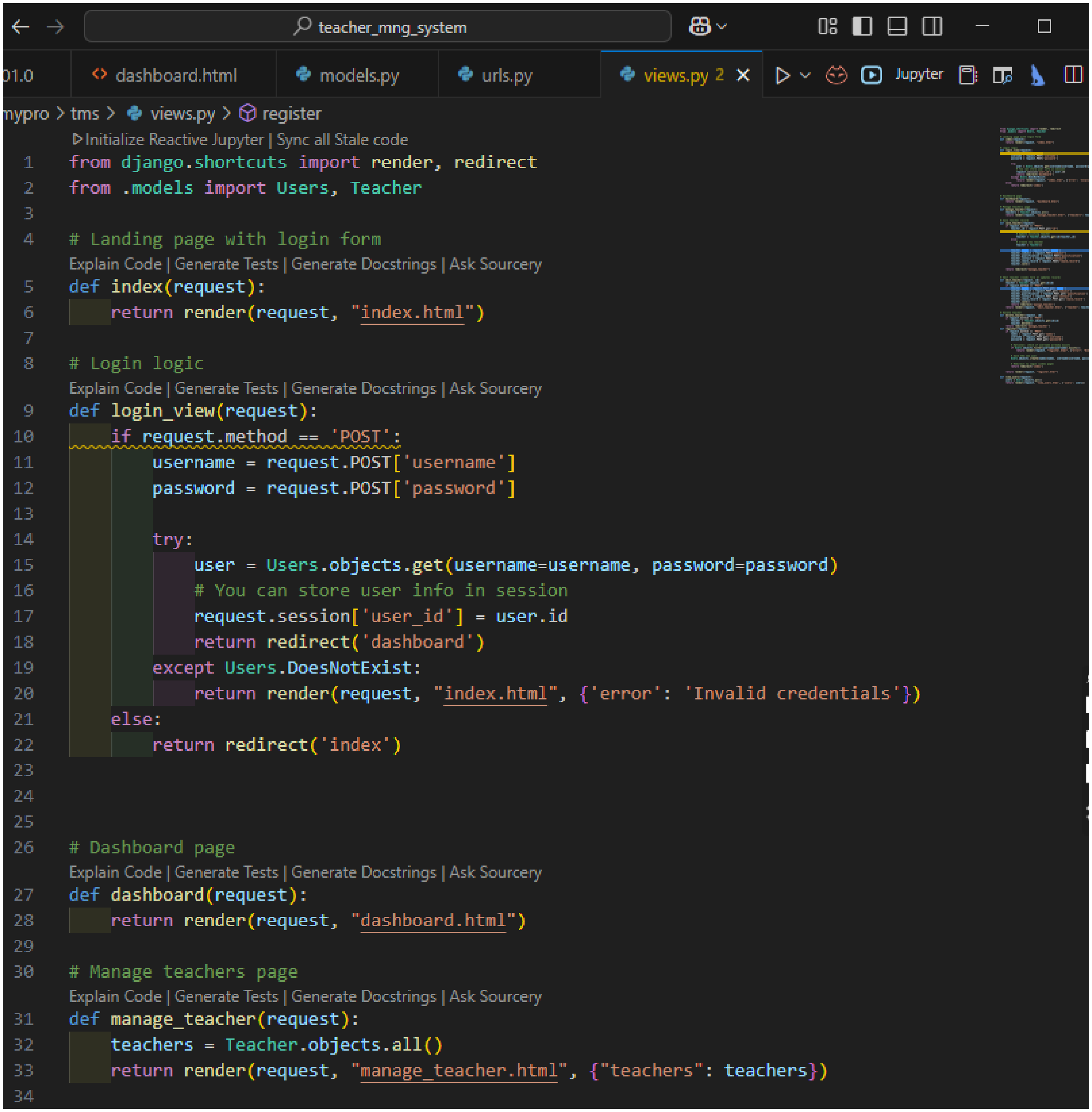
Username

Password

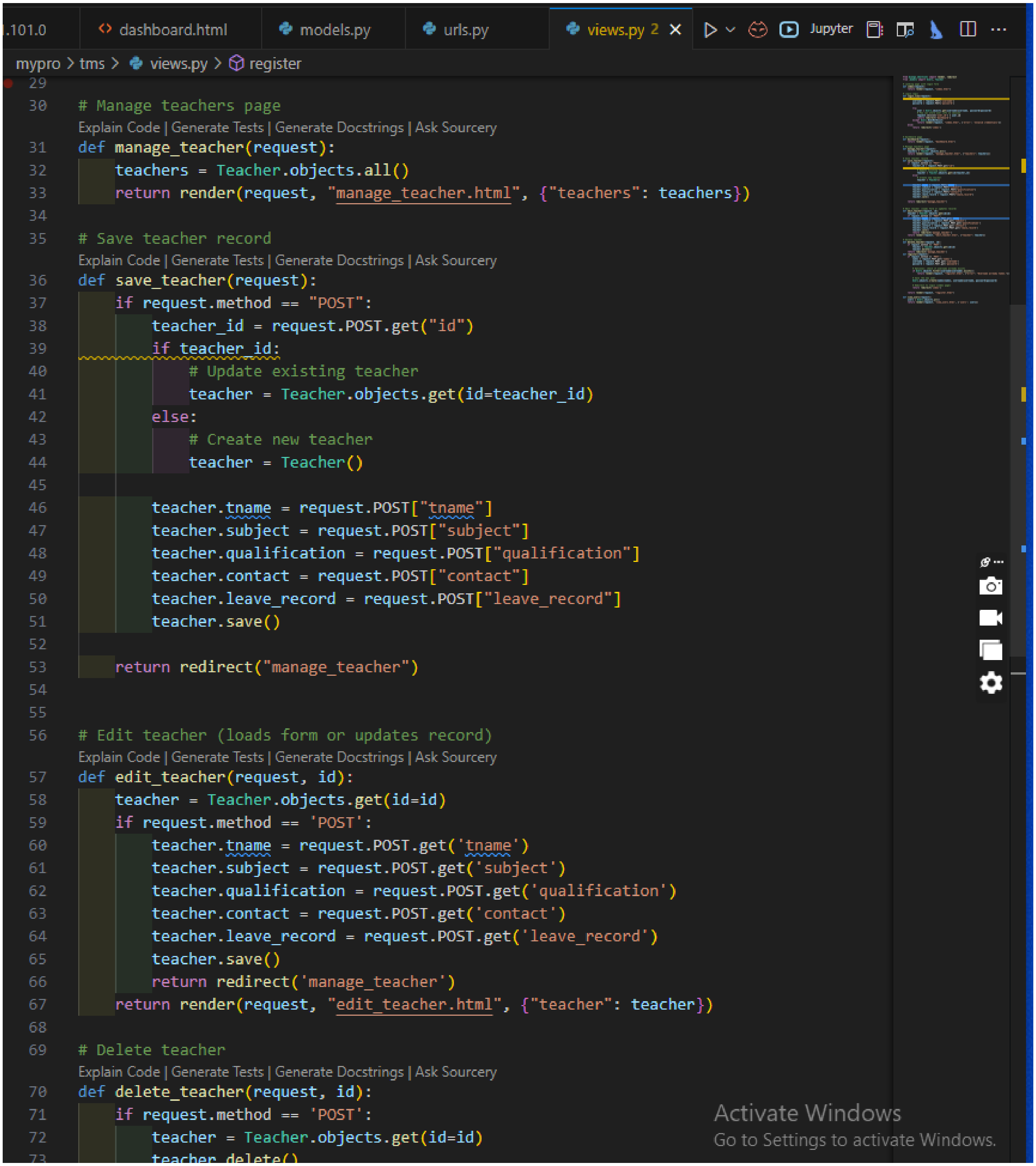


Release Notes: 1.101.0
dashboard.html
models.py
urls.py
Jupyter
mypro > tms > urls.py > ...
Initialize Reactive Jupyter | Sync all Stale code

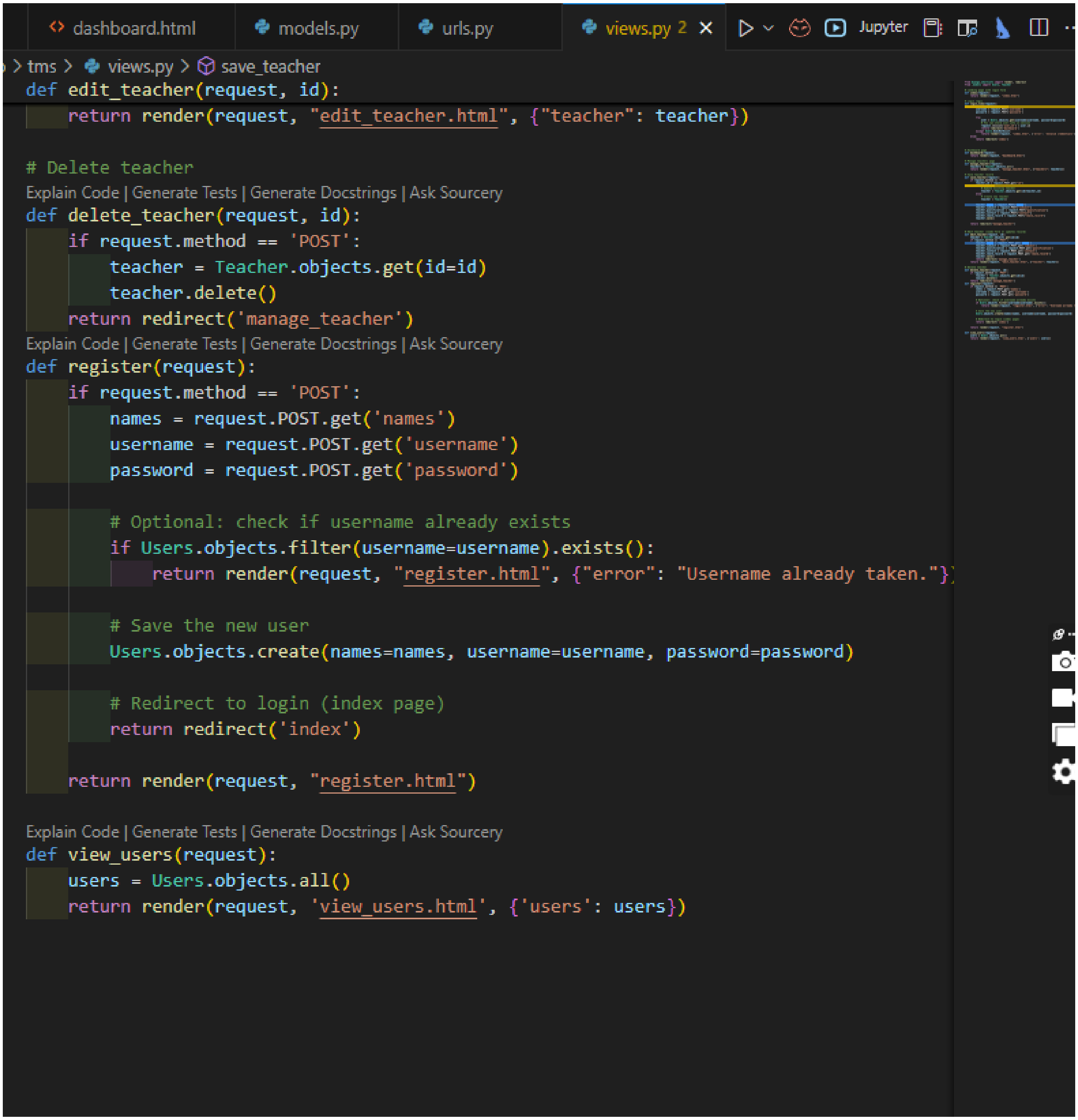
```python
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('login/', views.login_view, name='login'),
    path('register/', views.register, name='register'),
    path('dashboard/', views.dashboard, name='dashboard'),
    path('manage_teacher/', views.manage_teacher, name='manage_teacher'),
    path('save_teacher/', views.save_teacher, name='save_teacher'),
    path('edit_teacher/<int:id>/', views.edit_teacher, name='edit_teacher'),
    path('delete_teacher/<int:id>/', views.delete_teacher, name='delete_teacher'),
    path('view_users/', views.view_users, name='view_users'),
]
```


teacher_mng_system
01.0
dashboard.html
models.py
urls.py
views.py 2
Jupyter
nypro > tms > views.py > register
Initialize Reactive Jupyter | Sync all Stale code
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery

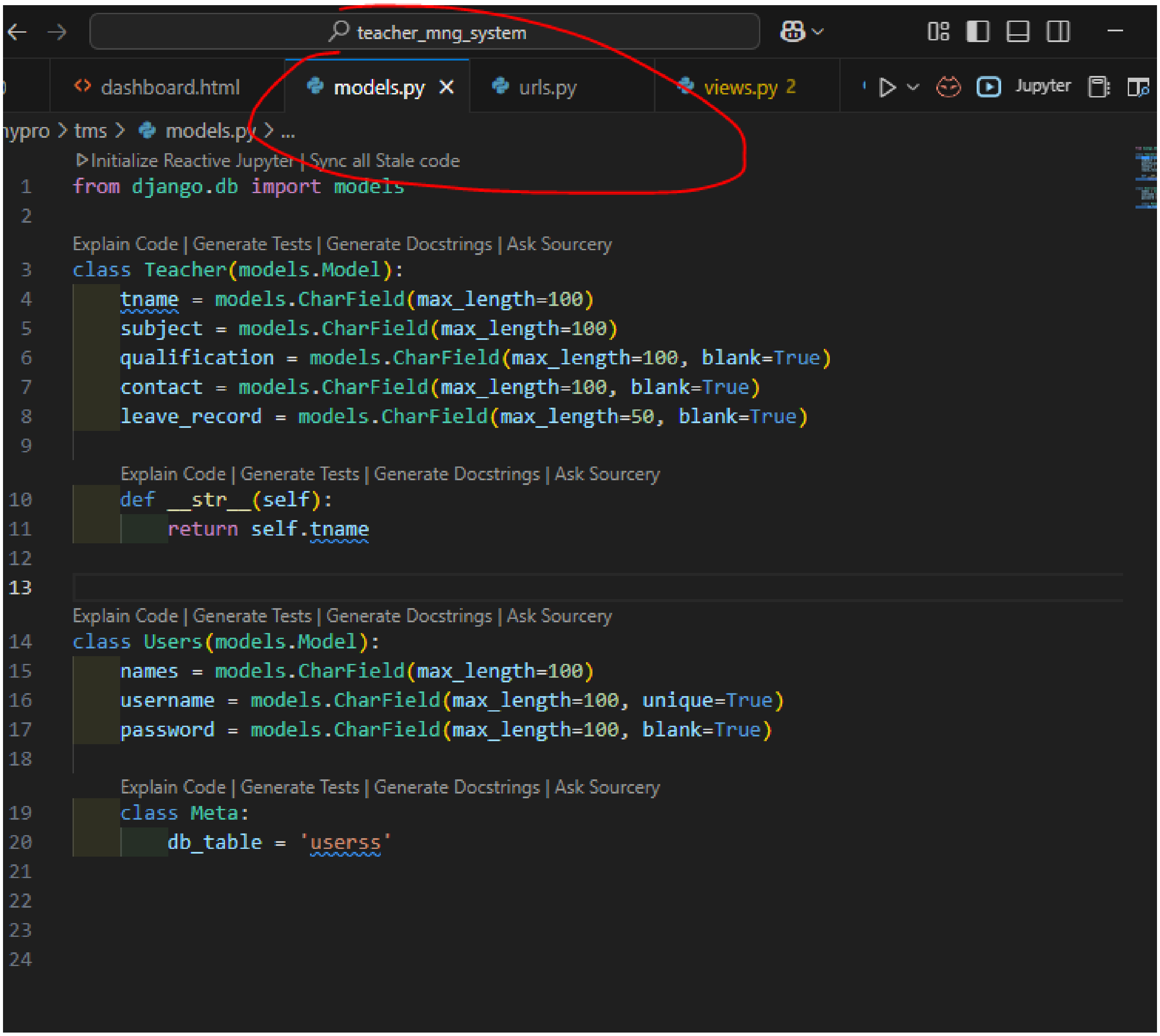
```python
from django.shortcuts import render, redirect
from .models import Users, Teacher

# Landing page with login form
def index(request):
    return render(request, "index.html")

# Login logic
def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']

        try:
            user = Users.objects.get(username=username, password=password)
            # You can store user info in session
            request.session['user_id'] = user.id
            return redirect('dashboard')
        except Users.DoesNotExist:
            return render(request, "index.html", {'error': 'Invalid credentials'})
    else:
        return redirect('index')



# Dashboard page
def dashboard(request):
    return render(request, "dashboard.html")

# Manage teachers page
def manage_teacher(request):
    teachers = Teacher.objects.all()
    return render(request, "manage_teacher.html", {"teachers": teachers})

```


.101.0
dashboard.html
models.py
urls.py
views.py 2
Jupyter
mypro > tms > views.py > register
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
Activate Windows
Go to Settings to activate Windows.

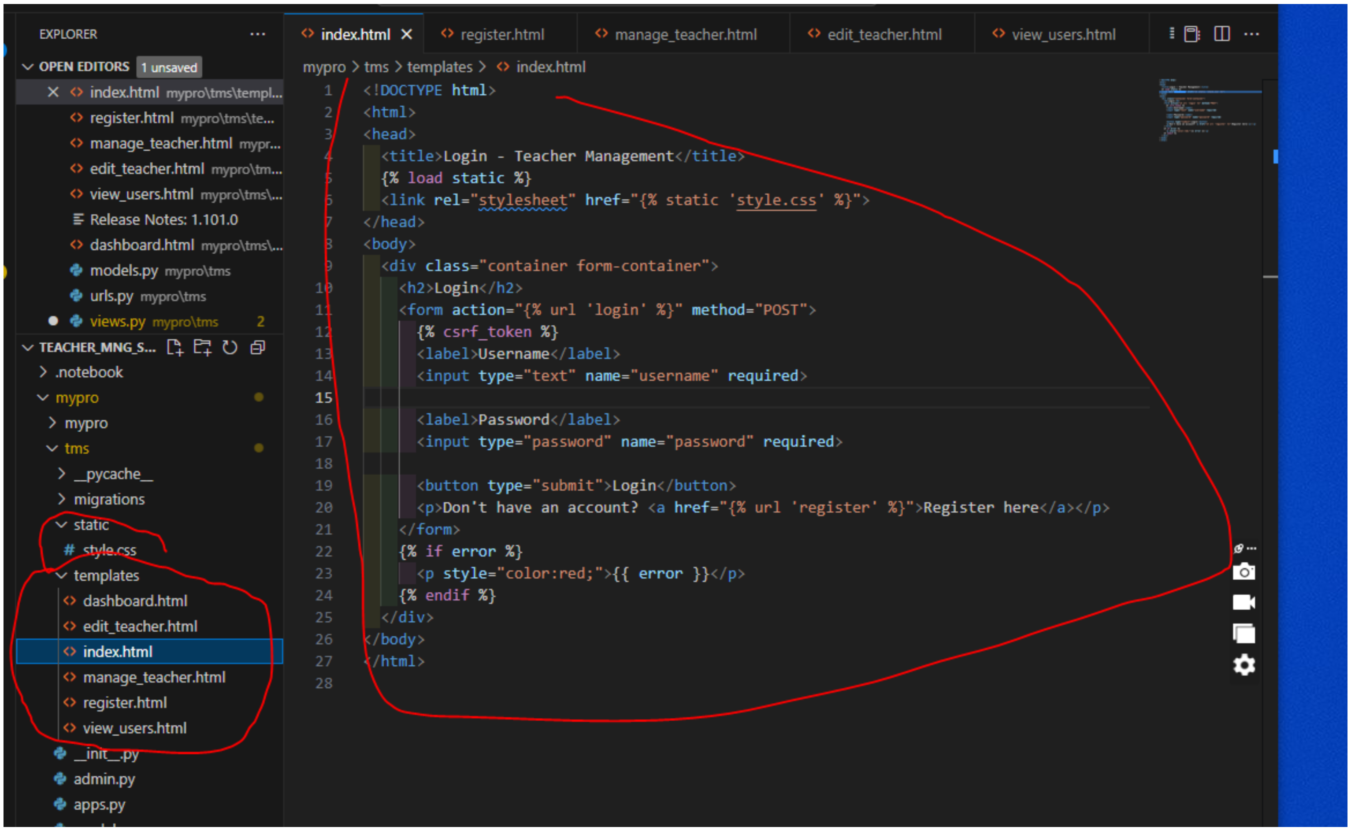
```python

# Manage teachers page
def manage_teacher(request):
    teachers = Teacher.objects.all()
    return render(request, "manage_teacher.html", {"teachers": teachers})

# Save teacher record
def save_teacher(request):
    if request.method == "POST":
        teacher_id = request.POST.get("id")
        if teacher_id:
            # Update existing teacher
            teacher = Teacher.objects.get(id=teacher_id)
        else:
            # Create new teacher
            teacher = Teacher()

        teacher.tname = request.POST["tname"]
        teacher.subject = request.POST["subject"]
        teacher.qualification = request.POST["qualification"]
        teacher.contact = request.POST["contact"]
        teacher.leave_record = request.POST["leave_record"]
        teacher.save()

    return redirect("manage_teacher")


# Edit teacher (loads form or updates record)
def edit_teacher(request, id):
    teacher = Teacher.objects.get(id=id)
    if request.method == 'POST':
        teacher.tname = request.POST.get('tname')
        teacher.subject = request.POST.get('subject')
        teacher.qualification = request.POST.get('qualification')
        teacher.contact = request.POST.get('contact')
        teacher.leave_record = request.POST.get('leave_record')
        teacher.save()
        return redirect('manage_teacher')
    return render(request, "edit_teacher.html", {"teacher": teacher})

# Delete teacher
def delete_teacher(request, id):
    if request.method == 'POST':
        teacher = Teacher.objects.get(id=id)
        teacher.delete()
```



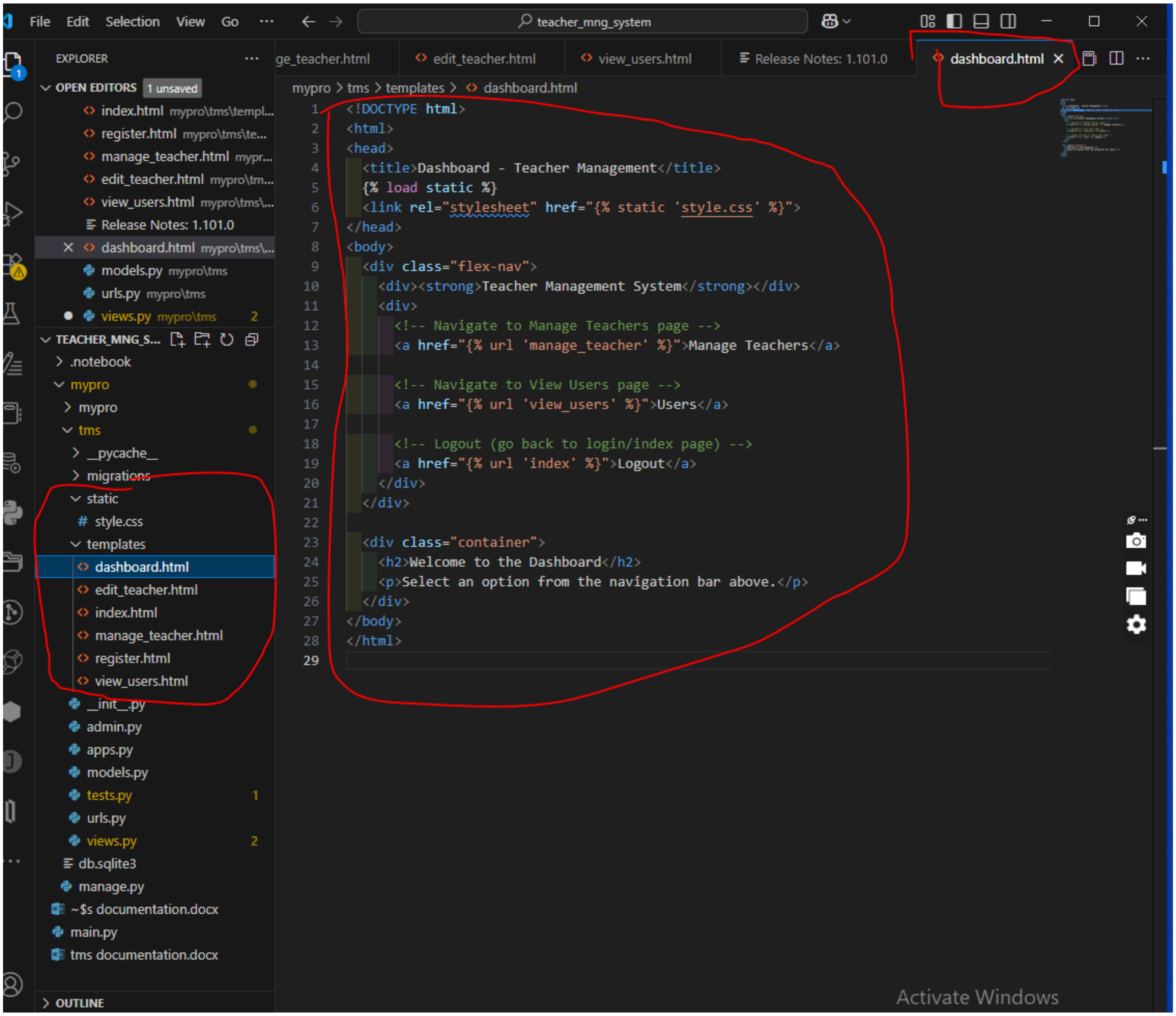
```
def edit_teacher(request, id):
    return render(request, "edit_teacher.html", {"teacher": teacher})

# Delete teacher
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
def delete_teacher(request, id):
    if request.method == 'POST':
        teacher = Teacher.objects.get(id=id)
        teacher.delete()
    return redirect('manage_teacher')
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
def register(request):
    if request.method == 'POST':
        names = request.POST.get('names')
        username = request.POST.get('username')
        password = request.POST.get('password')

        # Optional: check if username already exists
        if Users.objects.filter(username=username).exists():
            return render(request, "register.html", {"error": "Username already taken."}

        # Save the new user
        Users.objects.create(names=names, username=username, password=password)

        # Redirect to login (index page)
        return redirect('index')

    return render(request, "register.html")

Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery
def view_users(request):
    users = Users.objects.all()
    return render(request, 'view_users.html', {'users': users})
```


teacher_mng_system
dashboard.html
models.py
urls.py
views.py 2
Jupyter
ypro > tms > models.py > ...
▷ Initialize Reactive Jupyter | Sync all Stale code
Explain Code | Generate Tests | Generate Docstrings | Ask Sourcery

```
from django.db import models

class Teacher(models.Model):
    tname = models.CharField(max_length=100)
    subject = models.CharField(max_length=100)
    qualification = models.CharField(max_length=100, blank=True)
    contact = models.CharField(max_length=100, blank=True)
    leave_record = models.CharField(max_length=50, blank=True)

    def __str__(self):
        return self.tname


class Users(models.Model):
    names = models.CharField(max_length=100)
    username = models.CharField(max_length=100, unique=True)
    password = models.CharField(max_length=100, blank=True)

    class Meta:
        db_table = 'userss'
```



```html
<!DOCTYPE html>
<html>
<head>
    <title>Login - Teacher Management</title>
    {% load static %}
    <link rel="stylesheet" href="{% static 'style.css' %}">
</head>
<body>
    <div class="container form-container">
        <h2>Login</h2>
        <form action="{% url 'login' %}" method="POST">
            {% csrf_token %}
            <label>Username</label>
            <input type="text" name="username" required>

            <label>Password</label>
            <input type="password" name="password" required>

            <button type="submit">Login</button>
            <p>Don't have an account? <a href="{% url 'register' %}">Register here</a></p>
        </form>
        {% if error %}
            <p style="color:red;">{{ error }}</p>
        {% endif %}
    </div>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
    <title>Dashboard - Teacher Management</title>
    {% load static %}
    <link rel="stylesheet" href="{% static 'style.css' %}">
</head>
<body>
    <div class="flex-nav">
        <div><strong>Teacher Management System</strong></div>
        <div>
            <!-- Navigate to Manage Teachers page -->
            <a href="{% url 'manage_teacher' %}">Manage Teachers</a>

            <!-- Navigate to View Users page -->
            <a href="{% url 'view_users' %}">Users</a>

            <!-- Logout (go back to login/index page) -->
            <a href="{% url 'index' %}">Logout</a>
        </div>
    </div>

    <div class="container">
        <h2>Welcome to the Dashboard</h2>
        <p>Select an option from the navigation bar above.</p>
    </div>
</body>
</html>
```