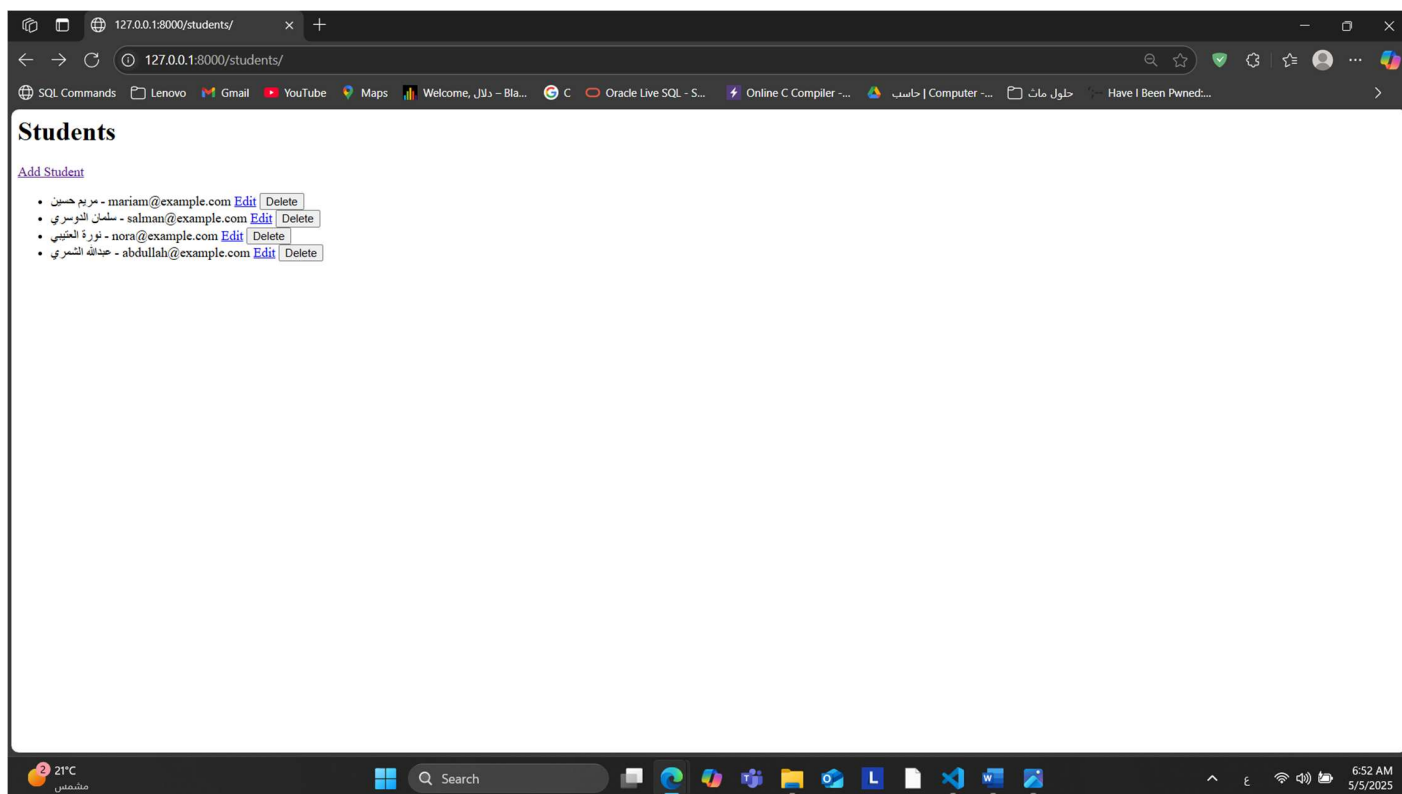
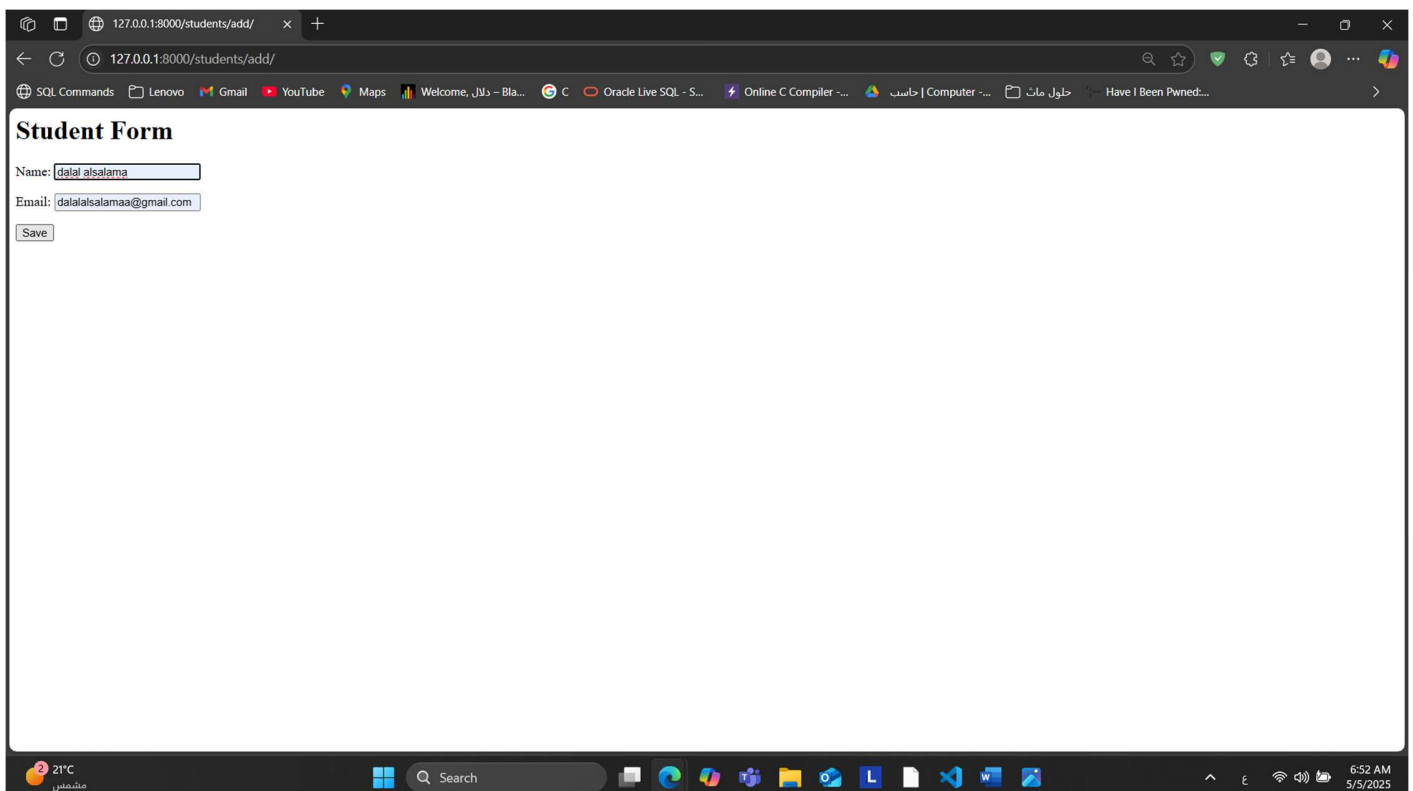
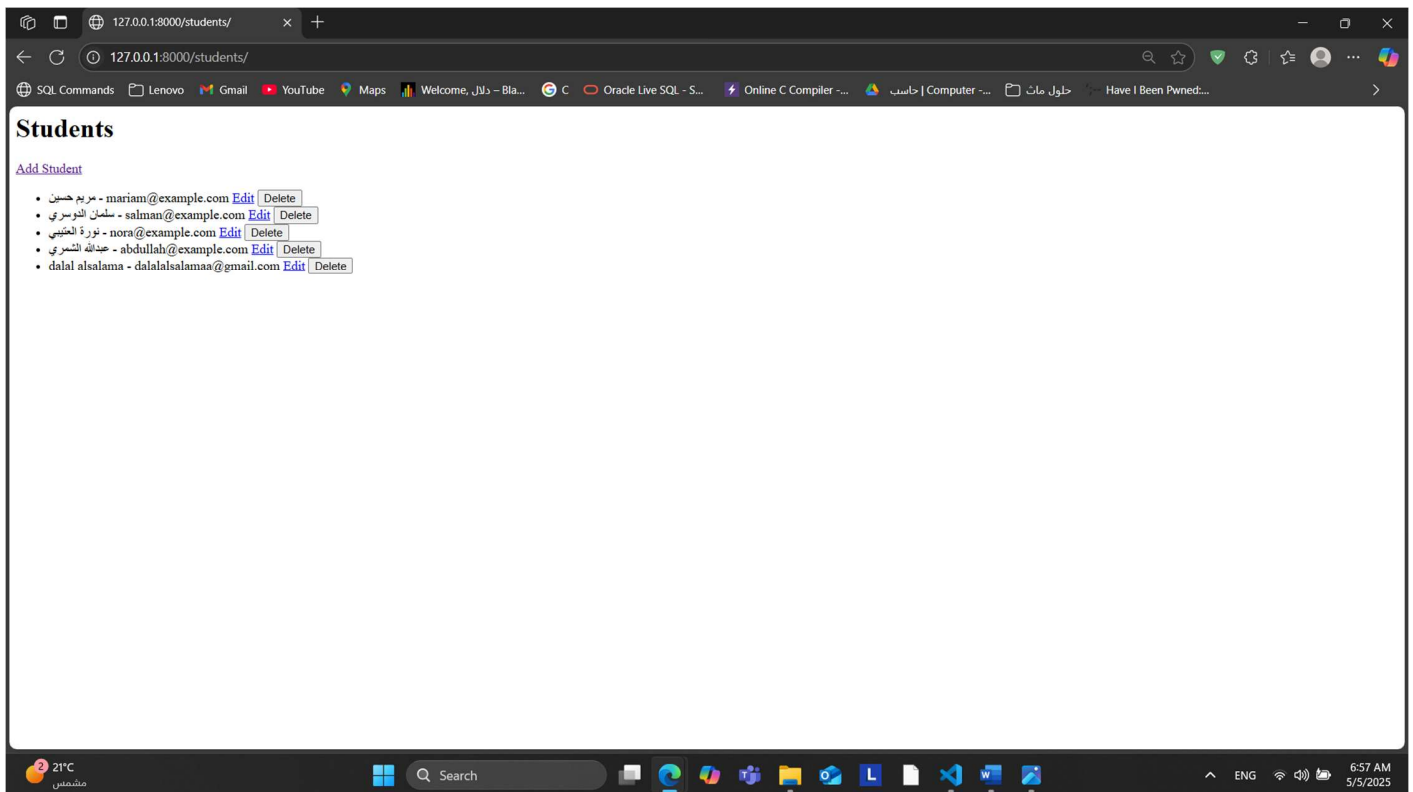


CS471 – Web Technologies (Laboratory)		Lab 11
		Django Forms (Part 2)

## Lab Activities:

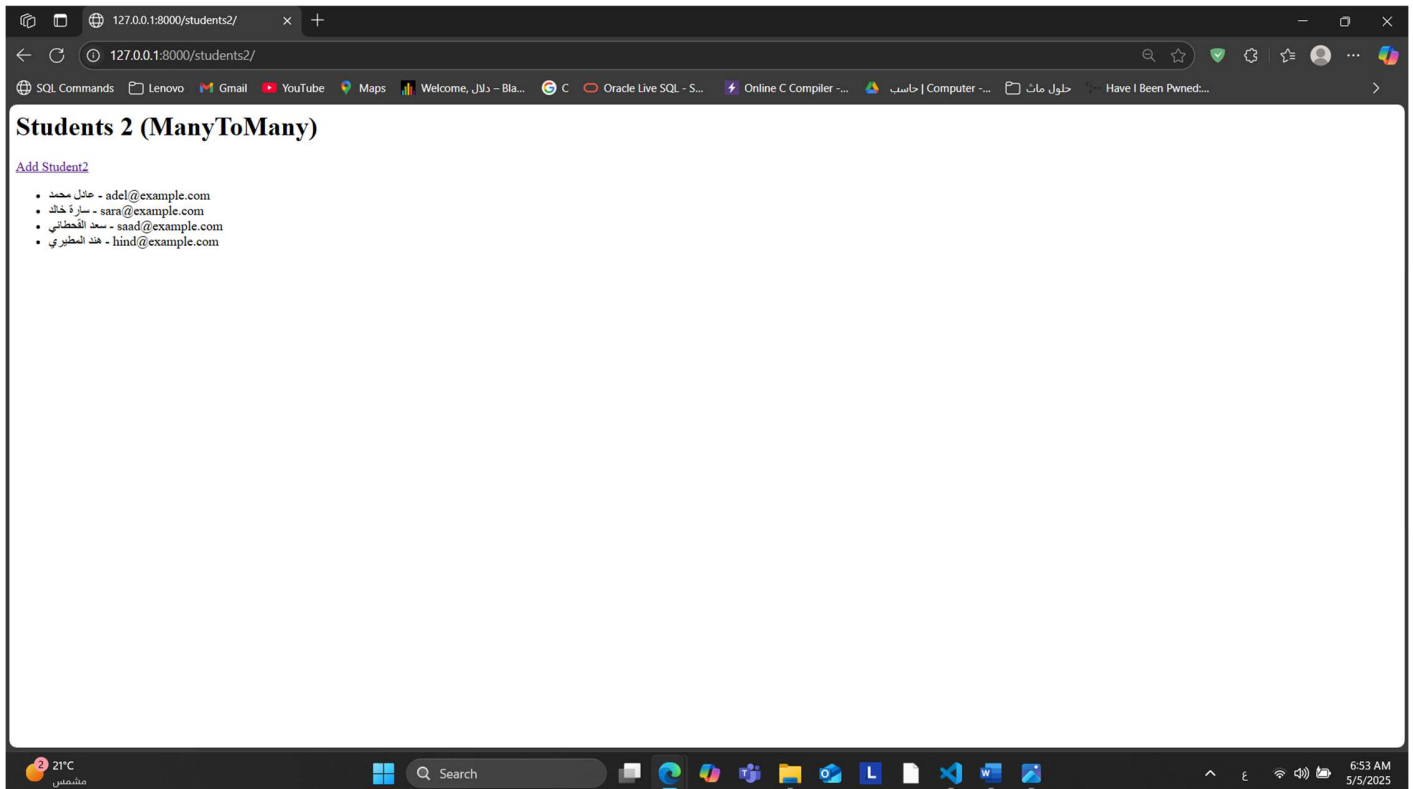
**Task 1:** Create a URL to list, add, update, or delete students and specify addresses (one address for each student) by applying the Django form field with any necessary HTML files and view functions.





**Task 2:** Suppose the relation between students and addresses is many-to-many, in which a student can have multiple addresses, and also, one address can be associated with many students. In this task, the code in the previous task should be copied and modified to implement many-to-many relationships.

**Note:** If needed, student or address models should be copied and renamed to, for example, student2 and address2.



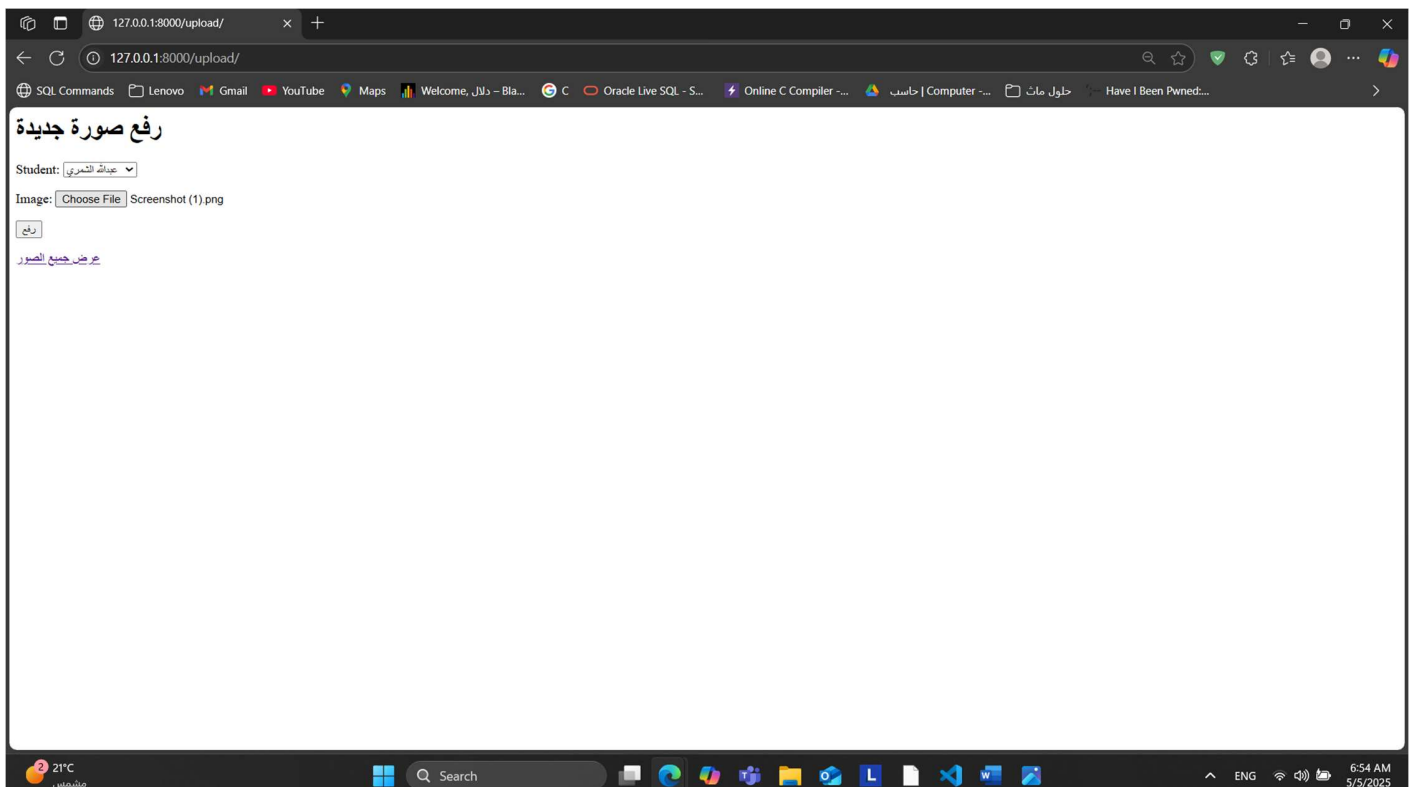
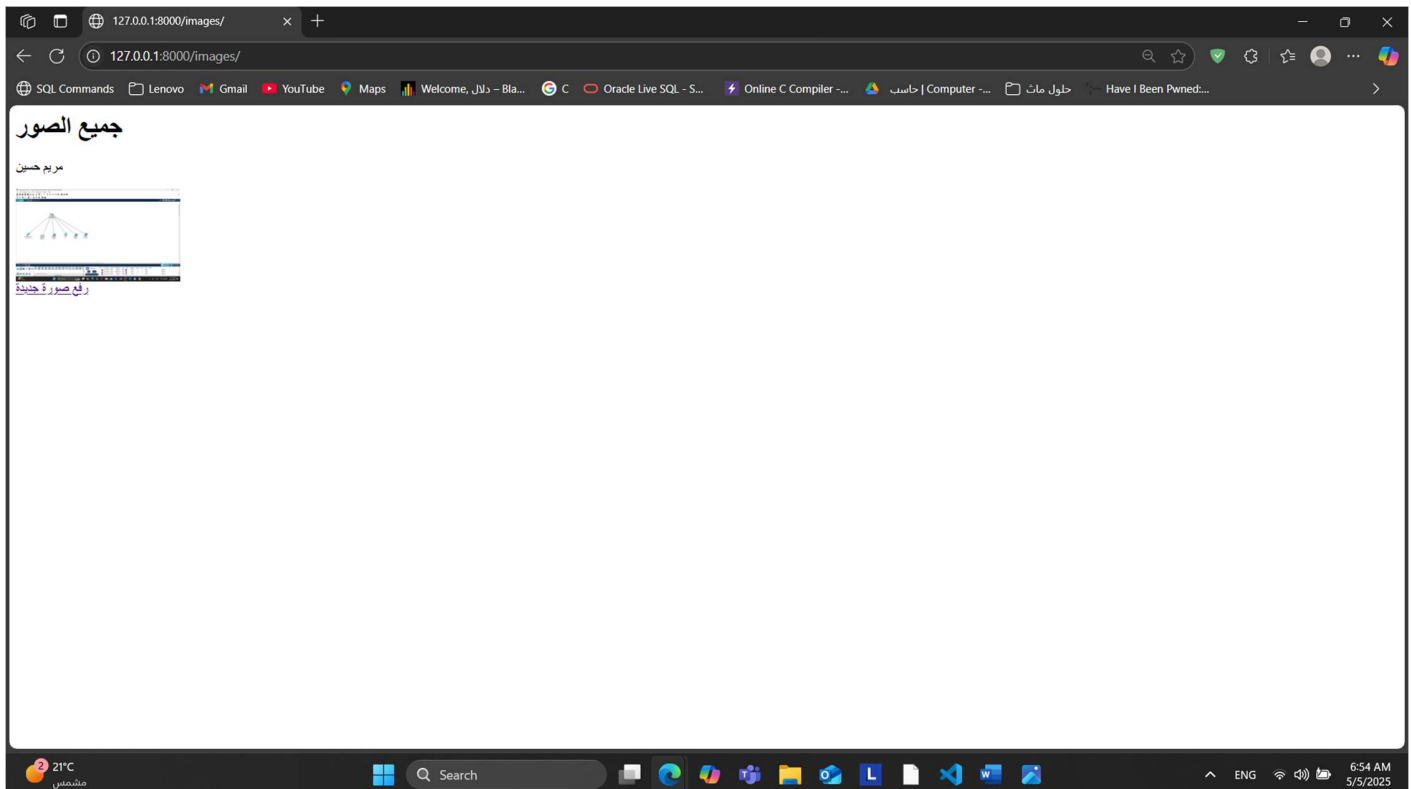
Student2 Form

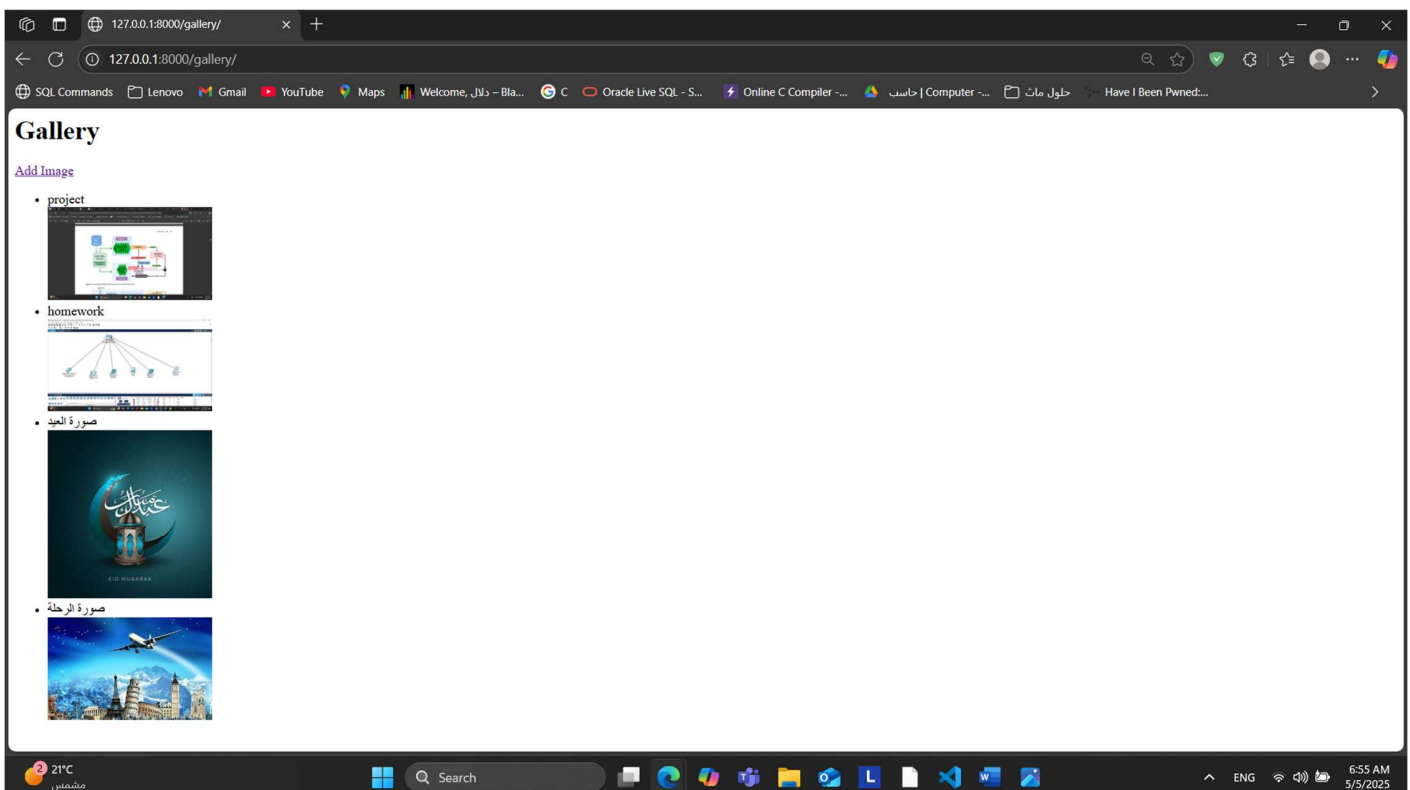
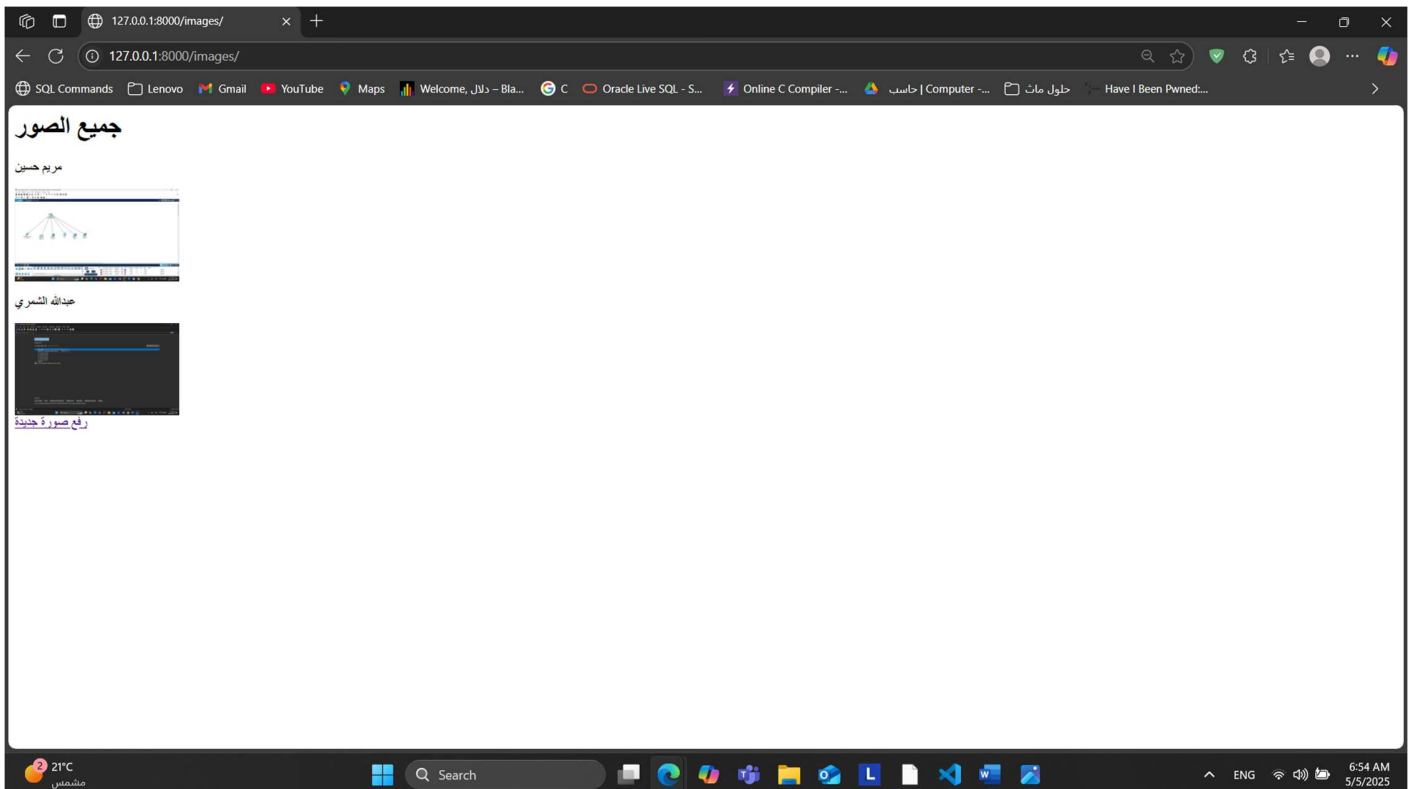
Name:

Email:

**Task 3:** Create an additional table (of your own choice) that includes images, and you should build forms to handle data with images through Django.

**Note:** A few configurations are needed for file handling.





```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Student, Address, Student2, Address2, Gallery
from .forms import StudentForm, AddressForm, Student2Form, Address2Form, GalleryForm

# Task 1 views
def student_list(request):
    students = Student.objects.all()
    return render(request, 'lab11/student_list.html', {'students': students})

def student_add(request):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            student = form.save()
            # إضافة العنوان مباشرة بعد إضافة الطالب
            address_form = AddressForm(request.POST)
            if address_form.is_valid():
                address = address_form.save(commit=False)
                address.student = student # ربط العنوان بالطالب
                address.save()
            return redirect('student_list')
    else:
        form = StudentForm()
        address_form = AddressForm()
    return render(request, 'lab11/student_form.html', {'form': form, 'address_form':
address_form})

def student_update(request, pk):
    student = get_object_or_404(Student, pk=pk)
    if request.method == 'POST':
        form = StudentForm(request.POST, instance=student)
        if form.is_valid():
            form.save()
            return redirect('student_list')
    else:
        form = StudentForm(instance=student)
    return render(request, 'lab11/student_form.html', {'form': form})

def student_delete(request, pk):
    student = get_object_or_404(Student, pk=pk)
    if request.method == 'POST':
        student.delete()
        return redirect('student_list')
    return render(request, 'lab11/student_confirm_delete.html', {'student': student})
```

```
# Task 2 views
def student2_list(request):
    students = Student2.objects.all()
    return render(request, 'lab11/student2_list.html', {'students': students})

def student2_add(request):
    if request.method == 'POST':
        form = Student2Form(request.POST)
        if form.is_valid():
            student = form.save()
            # إضافة العناوين المرتبطة
            address_form = Address2Form(request.POST)
            if address_form.is_valid():
                address = address_form.save(commit=False)
                address.students.add(student) # ربط الطالب بالعناوين
                address.save()
            return redirect('student2_list')
        else:
            form = Student2Form()
            address_form = Address2Form()
            return render(request, 'lab11/student2_form.html', {'form': form, 'address_form':
address_form})

# Task 3 views (Gallery)
def gallery_add(request):
    if request.method == 'POST':
        form = GalleryForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('gallery_list')
    else:
        form = GalleryForm()
    return render(request, 'lab11/gallery_form.html', {'form': form})

def gallery_list(request):
    galleries = Gallery.objects.all()
    return render(request, 'lab11/gallery_list.html', {'galleries': galleries})

from django.shortcuts import render, redirect
from .forms import StudentImageForm
from .models import StudentImage

def upload_image(request):
    if request.method == 'POST':
        form = StudentImageForm(request.POST, request.FILES)
```



```

        if form.is_valid():
            form.save()
            return redirect('image_list')
        else:
            form = StudentImageForm()
        return render(request, 'lab11/upload_image.html', {'form': form})

def image_list(request):
    images = StudentImage.objects.all()
    return render(request, 'lab11/image_list.html', {'images': images})

```

```

from django.urls import path
from . import views

urlpatterns = [
    # Task 1
    path('students/', views.student_list, name='student_list'),
    path('students/add/', views.student_add, name='student_add'),
    path('students/update/<int:pk>/', views.student_update, name='student_update'),
    path('students/delete/<int:pk>/', views.student_delete, name='student_delete'),

    # Task 2
    path('students2/', views.student2_list, name='student2_list'),
    path('students2/add/', views.student2_add, name='student2_add'),

    # Task 3
    path('gallery/', views.gallery_list, name='gallery_list'),
    path('gallery/add/', views.gallery_add, name='gallery_add'),

    path('upload/', views.upload_image, name='upload_image'),
    path('images/', views.image_list, name='image_list'),
]

```

```
from django.db import models

# Task 1: OneToOne Relation
class Student(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Address(models.Model):
    student = models.OneToOneField(Student, on_delete=models.CASCADE)
    city = models.CharField(max_length=100)
    street = models.CharField(max_length=100)

    def __str__(self):
        return f"{self.city}, {self.street}"

# Task 2: ManyToMany Relation
class Student2(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Address2(models.Model):
    students = models.ManyToManyField(Student2)
    city = models.CharField(max_length=100)
    street = models.CharField(max_length=100)

    def __str__(self):
        return f"{self.city}, {self.street}"

# Task 3: Table with Images
class Gallery(models.Model):
    title = models.CharField(max_length=100)
    image = models.ImageField(upload_to='images/')

    def __str__(self):
        return self.title

class StudentImage(models.Model):
```

```
student = models.ForeignKey(Student, on_delete=models.CASCADE, related_name='images')
image = models.ImageField(upload_to='student_images/')

def __str__(self):
    return f"صورة للطالب: {self.student.name}"
```