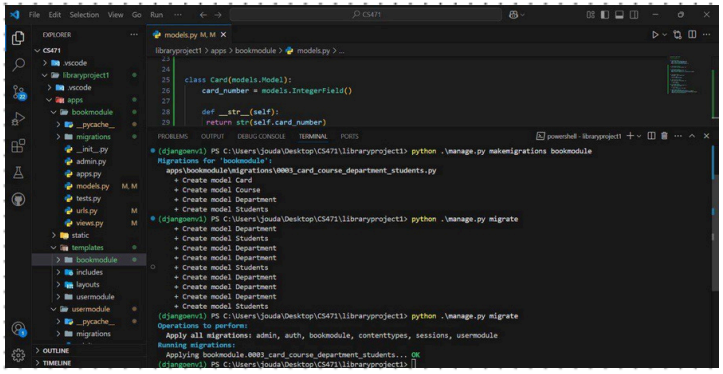
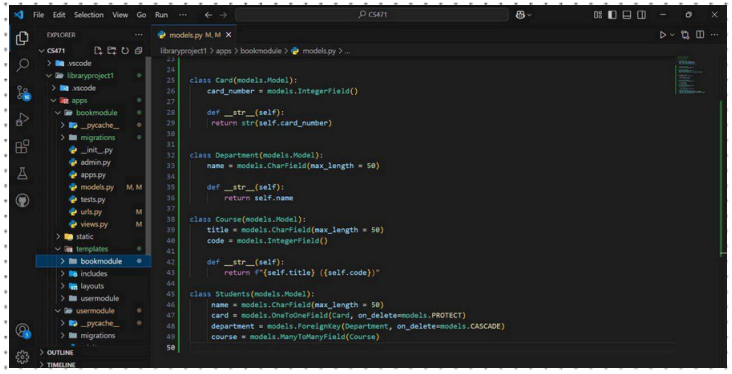
Lab 9

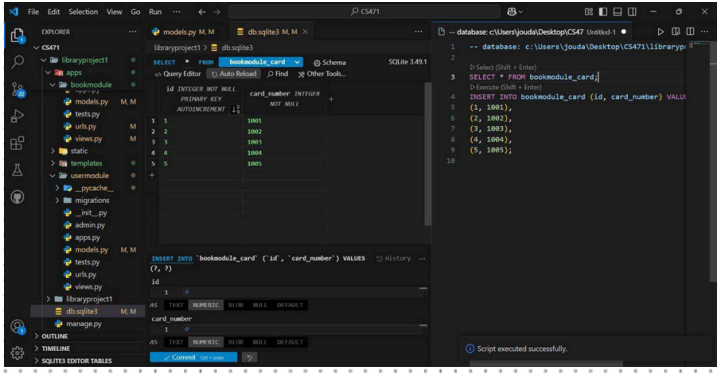
Django Models (Part 3)

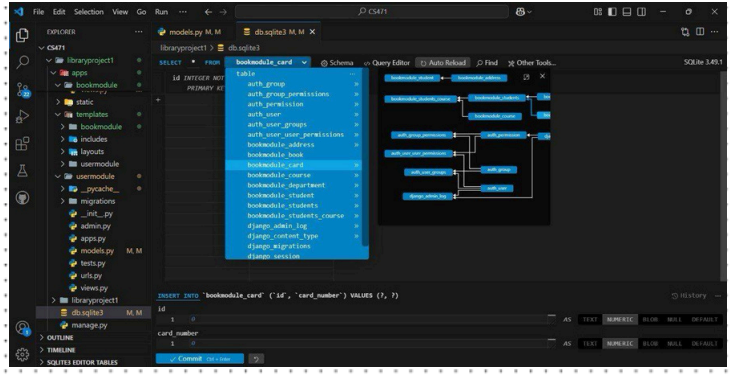
Student Name: Joud Alblaihed
University ID: 422205657
Subject: CS 471
Section number: 3526
Doctor Name: Bshayer Alkhalifa

Part 1, 2 and 3

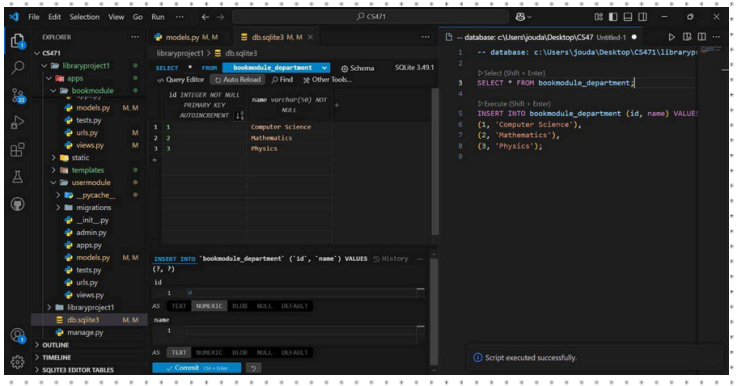


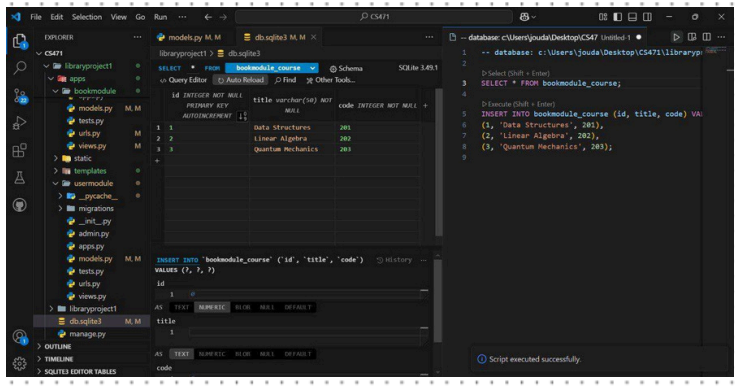


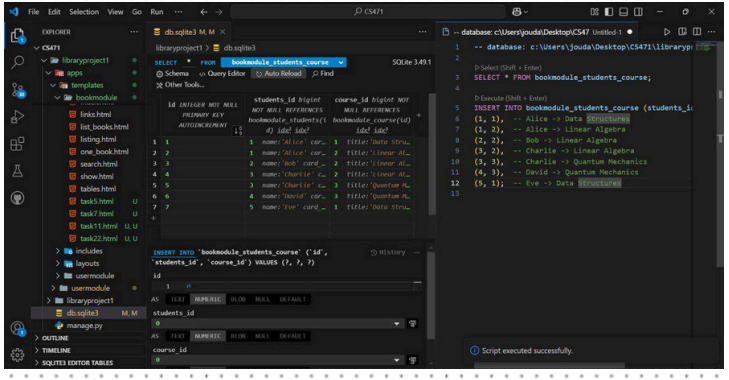


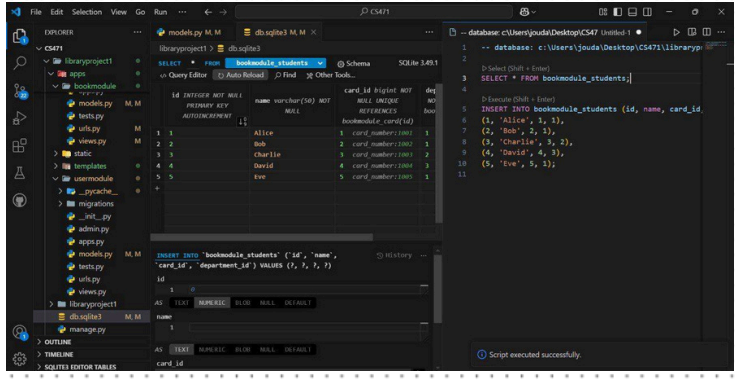


Part 1, 2 and 3

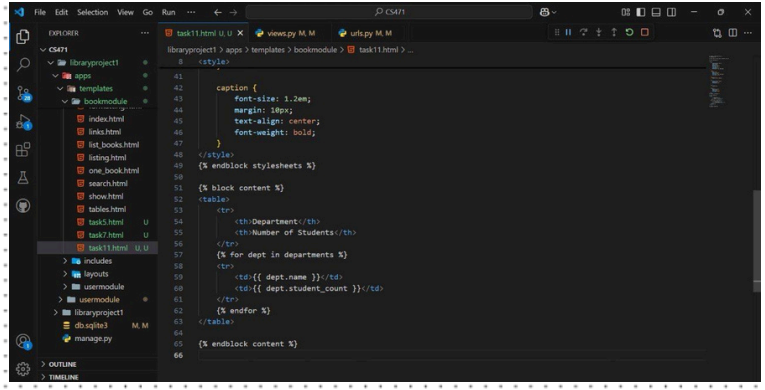


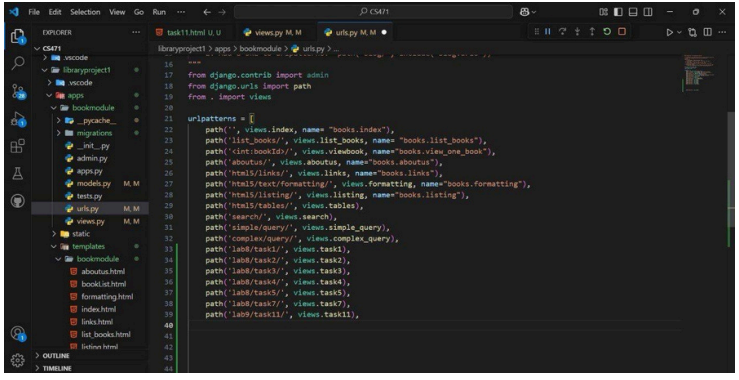


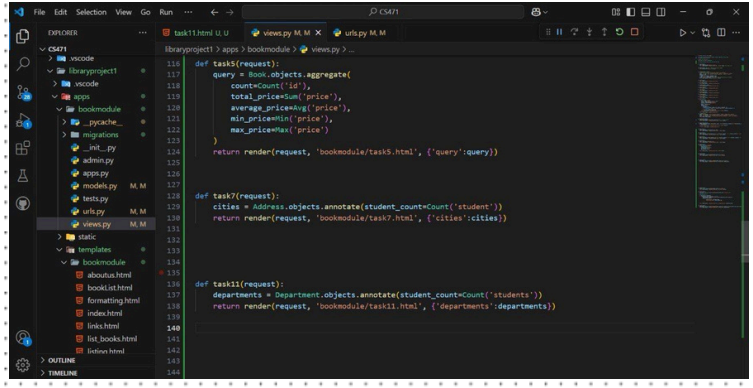




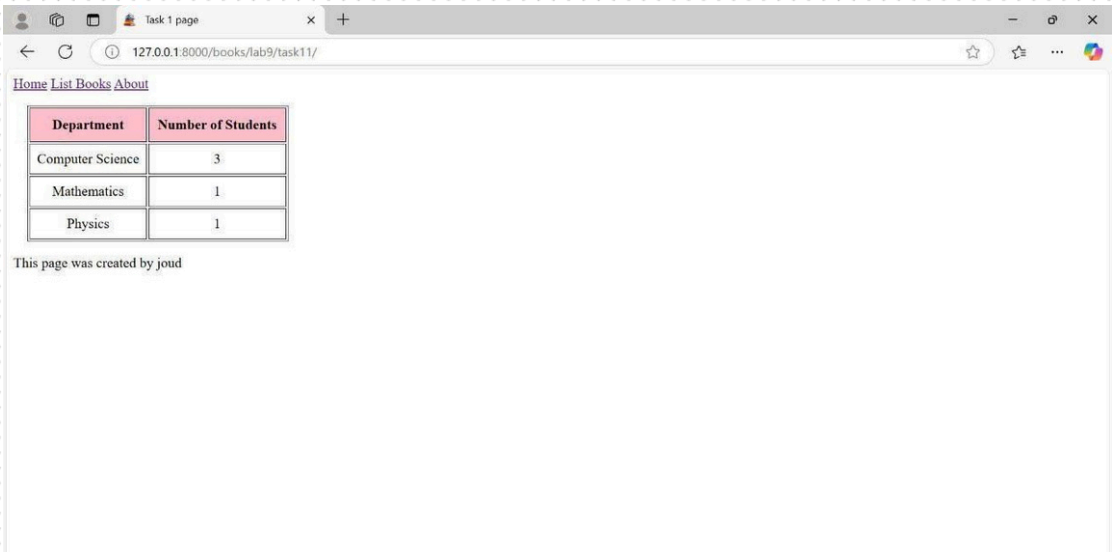
Task 1
Code


task11.html
<style>
caption {
font-size: 1.2em;
margin: 10px;
text-align: center;
font-weight: bold;
}
</style>
{% endblock stylesheets %}
{% block content %}
<table>
<tr>
<th>Department</th>
<th>Number of Students</th>
</tr>
{% for dept in departments %}
<tr>
<td>{{ dept.name }}</td>
<td>{{ dept.student_count }}</td>
</tr>
{% endfor %}
</table>
{% endblock content %}

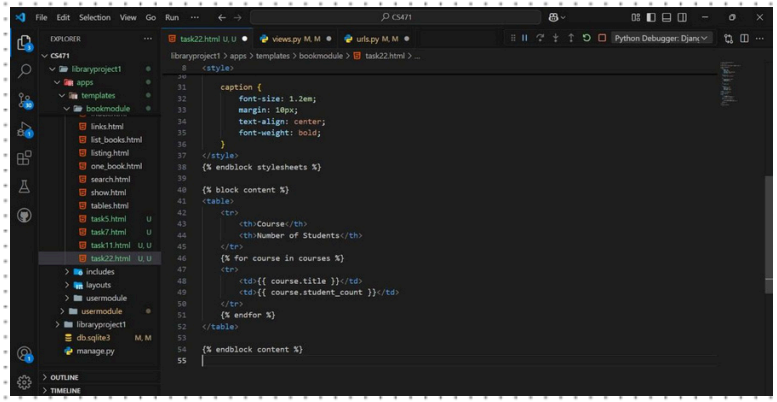

from django.contrib import admin
from django.urls import path
from . import views
urlpatterns = [
path('', views.index, name= "books.index"),
path('list_books/', views.list_books, name= "books.list_books"),
path('<int:bookId>/', views.viewbook, name="books.view_one_book"),
path('aboutus/', views.aboutus, name="books.aboutus"),
path('html5/links/', views.links, name="books.links"),
path('html5/text/formatting/', views.formatting, name="books.formatting"),
path('html5/listing/', views.listing, name="books.listing"),
path('html5/tables/', views.tables),
path('search/', views.search),
path('simple/query/', views.simple_query),
path('complex/query/', views.complex_query),
path('lab8/task1/', views.task1),
path('lab8/task2/', views.task2),
path('lab8/task3/', views.task3),
path('lab8/task4/', views.task4),
path('lab8/task5/', views.task5),
path('lab8/task7/', views.task7),
path('lab9/task11/', views.task11),


def task5(request):
query = Book.objects.aggregate(
count=Count('id'),
total_price=Sum('price'),
average_price=Avg('price'),
min_price=Min('price'),
max_price=Max('price')
)
return render(request, 'bookmodule/task5.html', {'query':query})
def task7(request):
cities = Address.objects.annotate(student_count=Count('student'))
return render(request, 'bookmodule/task7.html', {'cities':cities})
def task11(request):
departments = Department.objects.annotate(student_count=Count('students'))
return render(request, 'bookmodule/task11.html', {'departments':departments})

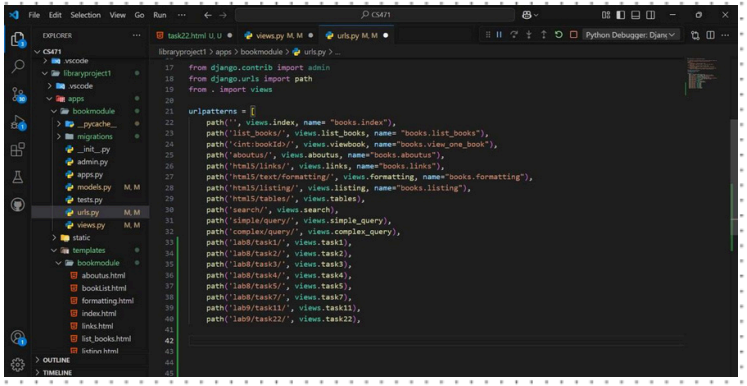
Task 1
Output



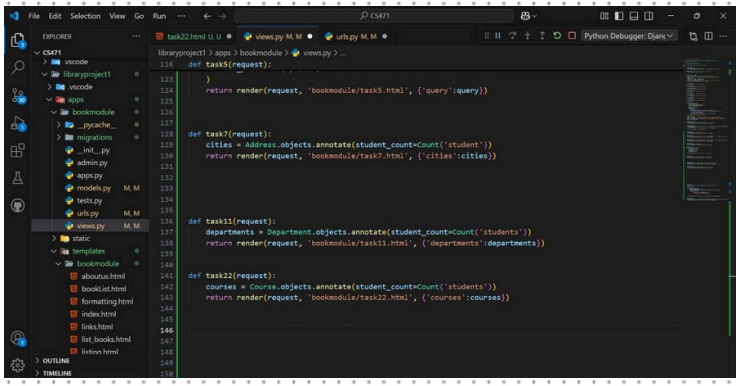
Task 2
Code



task22.html U, U
views.py M, M
urls.py M, M
Python Debugger: Djang
libraryproject1 > apps > templates > bookmodule > task22.html > ...
<style>
caption {
font-size: 1.2em;
margin: 10px;
text-align: center;
font-weight: bold;
}
</style>
{% endblock stylesheets %}
{% block content %}
<table>
<tr>
<th>Course</th>
<th>Number of Students</th>
</tr>
{% for course in courses %}
<tr>
<td>{{ course.title }}</td>
<td>{{ course.student_count }}</td>
</tr>
{% endfor %}
</table>
{% endblock content %}

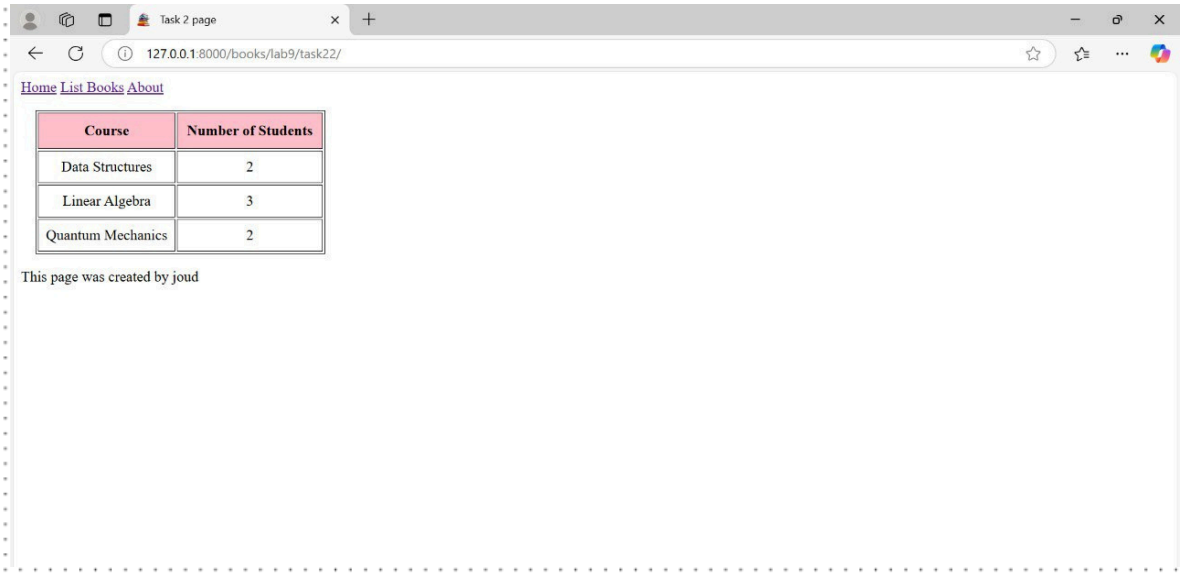


from django.contrib import admin
from django.urls import path
from . import views
urlpatterns = [
path('', views.index, name= "books.index"),
path('list_books/', views.list_books, name= "books.list_books"),
path('<int:bookId>/', views.viewbook, name="books.view_one_book"),
path('aboutus/', views.aboutus, name="books.aboutus"),
path('html5/links/', views.links, name="books.links"),
path('html5/text/formatting/', views.formatting, name="books.formatting"),
path('html5/listing/', views.listing, name="books.listing"),
path('html5/tables/', views.tables),
path('search/', views.search),
path('simple/query/', views.simple_query),
path('complex/query/', views.complex_query),
path('lab8/task1/', views.task1),
path('lab8/task2/', views.task2),
path('lab8/task3/', views.task3),
path('lab8/task4/', views.task4),
path('lab8/task5/', views.task5),
path('lab8/task7/', views.task7),
path('lab9/task11/', views.task11),
path('lab9/task22/', views.task22),

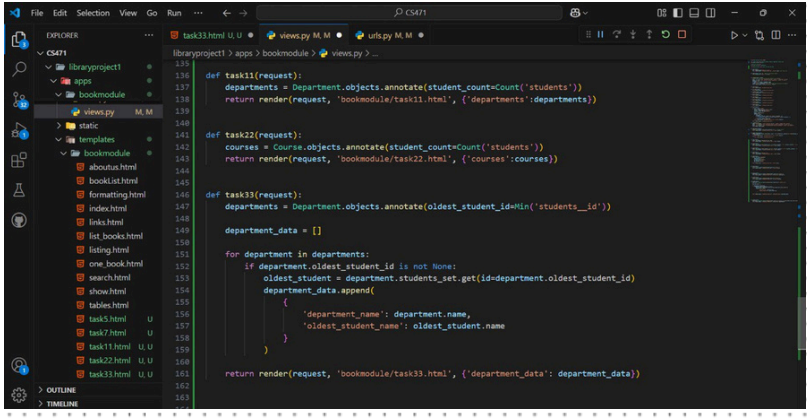


def task5(request):
)
return render(request, 'bookmodule/task5.html', {'query':query})
def task7(request):
cities = Address.objects.annotate(student_count=Count('student'))
return render(request, 'bookmodule/task7.html', {'cities':cities})
def task11(request):
departments = Department.objects.annotate(student_count=Count('students'))
return render(request, 'bookmodule/task11.html', {'departments':departments})
def task22(request):
courses = Course.objects.annotate(student_count=Count('students'))
return render(request, 'bookmodule/task22.html', {'courses':courses})

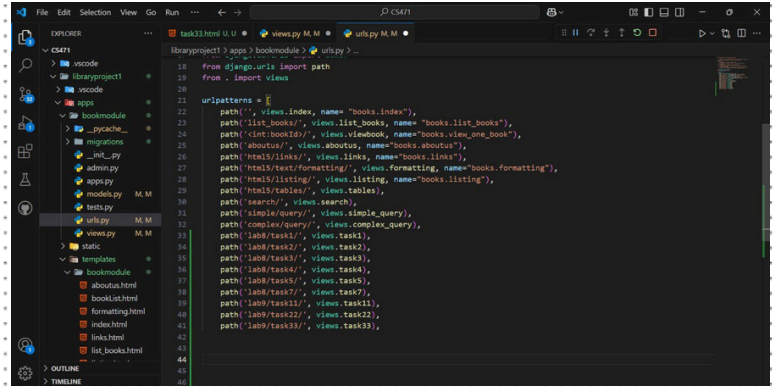
Task 2
Output



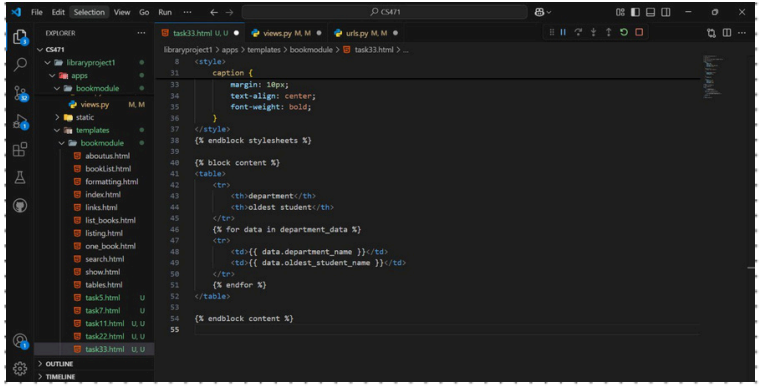
Task 3
Code



```python
def task11(request):
    departments = Department.objects.annotate(student_count=Count('students'))
    return render(request, 'bookmodule/task11.html', {'departments':departments})


def task22(request):
    courses = Course.objects.annotate(student_count=Count('students'))
    return render(request, 'bookmodule/task22.html', {'courses':courses})


def task33(request):
    departments = Department.objects.annotate(oldest_student_id=Min('students__id'))

    department_data = []

    for department in departments:
        if department.oldest_student_id is not None:
            oldest_student = department.students_set.get(id=department.oldest_student_id)
            department_data.append(
                {
                    'department_name': department.name,
                    'oldest_student_name': oldest_student.name
                }
            )

    return render(request, 'bookmodule/task33.html', {'department_data': department_data})
```

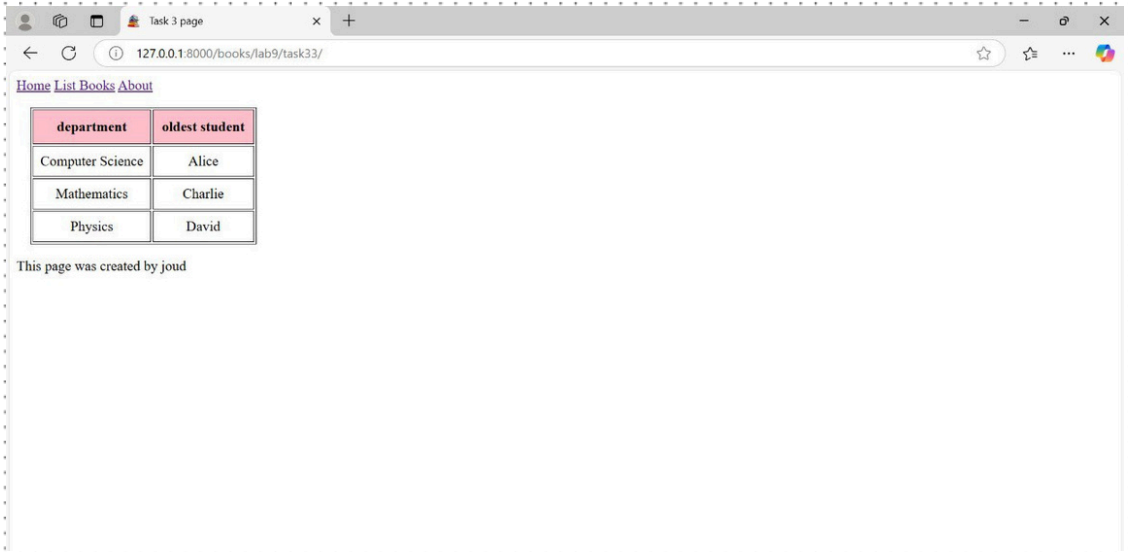


```python
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name= "books.index"),
    path('list_books/', views.list_books, name= "books.list_books"),
    path('<int:bookId>/', views.viewbook, name="books.view_one_book"),
    path('aboutus/', views.aboutus, name="books.aboutus"),
    path('html5/links/', views.links, name="books.links"),
    path('html5/text/formatting/', views.formatting, name="books.formatting"),
    path('html5/listing/', views.listing, name="books.listing"),
    path('html5/tables/', views.tables),
    path('search/', views.search),
    path('simple/query/', views.simple_query),
    path('complex/query/', views.complex_query),
    path('lab8/task1/', views.task1),
    path('lab8/task2/', views.task2),
    path('lab8/task3/', views.task3),
    path('lab8/task4/', views.task4),
    path('lab8/task5/', views.task5),
    path('lab8/task7/', views.task7),
    path('lab9/task11/', views.task11),
    path('lab9/task22/', views.task22),
    path('lab9/task33/', views.task33),
```

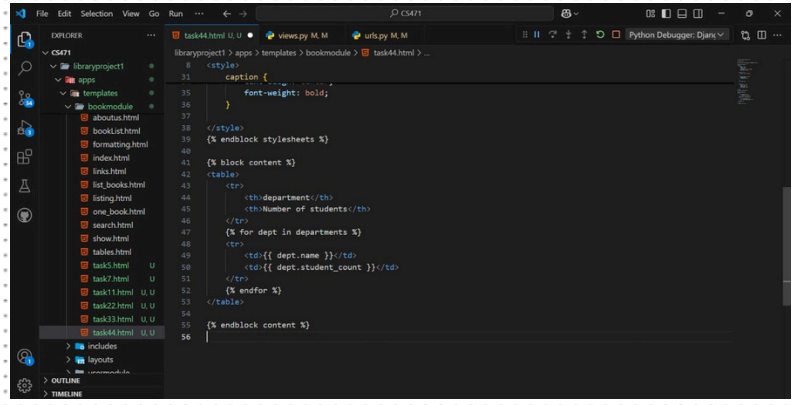


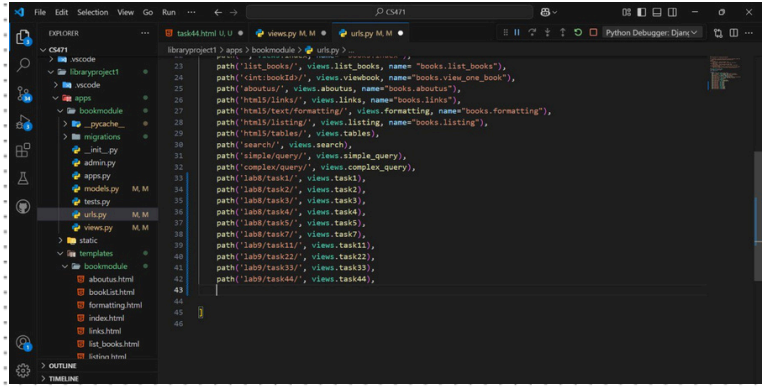
```
<style>
    caption {
        margin: 10px;
        text-align: center;
        font-weight: bold;
    }
</style>
{% endblock stylesheets %}

{% block content %}
<table>
    <tr>
        <th>department</th>
        <th>oldest student</th>
    </tr>
    {% for data in department_data %}
    <tr>
        <td>{{ data.department_name }}</td>
        <td>{{ data.oldest_student_name }}</td>
    </tr>
    {% endfor %}
</table>

{% endblock content %}
```

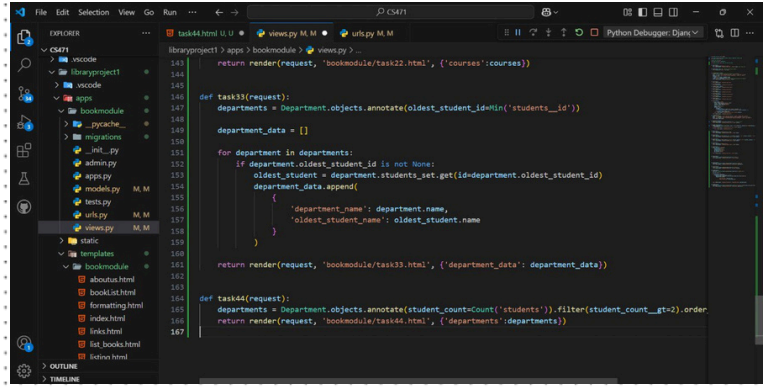
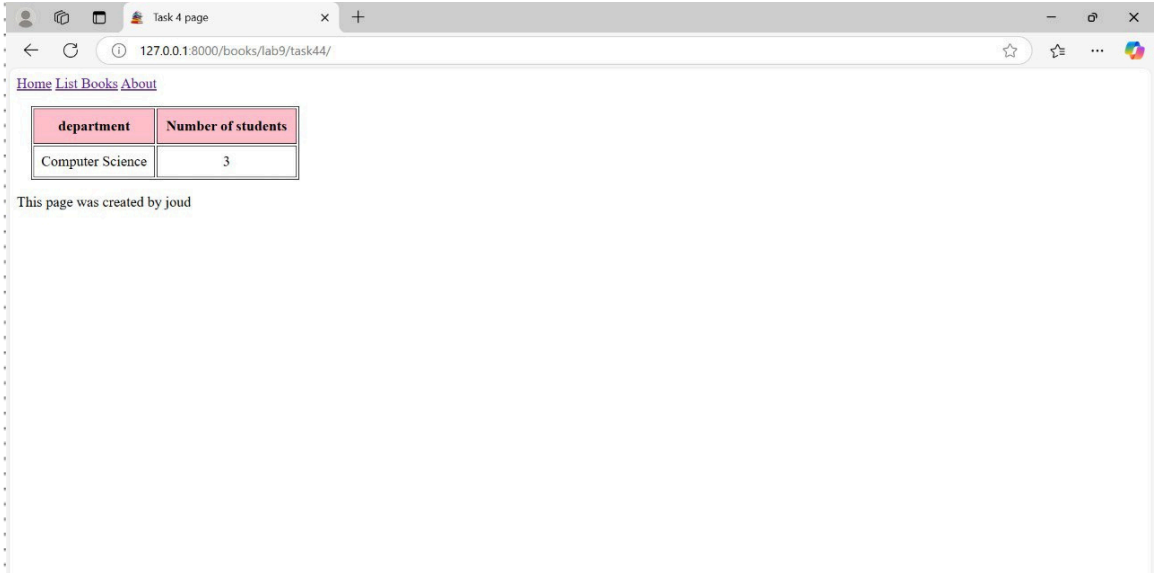
Task 3
Output



Task 4
Code


task44.html
<style>
caption {
font-weight: bold;
}
</style>
{% endblock stylesheets %}
{% block content %}
<table>
<tr>
<th>department</th>
<th>Number of students</th>
</tr>
{% for dept in departments %}
<tr>
<td>{{ dept.name }}</td>
<td>{{ dept.student_count }}</td>
</tr>
{% endfor %}
</table>
{% endblock content %}


urls.py
path('list_books/', views.list_books, name= "books.list_books"),
path('<int:bookId>/', views.viewbook, name="books.view_one_book"),
path('aboutus/', views.aboutus, name="books.aboutus"),
path('html5/links/', views.links, name="books.links"),
path('html5/text/formatting/', views.formatting, name="books.formatting"),
path('html5/listing/', views.listing, name="books.listing"),
path('html5/tables/', views.tables),
path('search/', views.search),
path('simple/query/', views.simple_query),
path('complex/query/', views.complex_query),
path('lab8/task1/', views.task1),
path('lab8/task2/', views.task2),
path('lab8/task3/', views.task3),
path('lab8/task4/', views.task4),
path('lab8/task5/', views.task5),
path('lab8/task7/', views.task7),
path('lab9/task11/', views.task11),
path('lab9/task22/', views.task22),
path('lab9/task33/', views.task33),
path('lab9/task44/', views.task44),


views.py
return render(request, 'bookmodule/task22.html', {'courses':courses})
def task33(request):
departments = Department.objects.annotate(oldest_student_id=Min('students__id'))
department_data = []
for department in departments:
if department.oldest_student_id is not None:
oldest_student = department.students_set.get(id=department.oldest_student_id)
department_data.append(
{
'department_name': department.name,
'oldest_student_name': oldest_student.name
}
)
return render(request, 'bookmodule/task33.html', {'department_data': department_data})
def task44(request):
departments = Department.objects.annotate(student_count=Count('students')).filter(student_count__gt=2).order
return render(request, 'bookmodule/task44.html', {'departments':departments})

Task 4
Output