**національний університет БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ україни**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра комп’ютерних наук

**ЛАБОРАТОРНА РОБОТА №8**

Виконав:

студент групи ІПЗ-21007б

Щур В. М.

Київ – 2024

**Хід виконання**

*1. На платформі Docker, за допомогою файлу docker-compose.yml, створити контейнер в якому створити Django-застосунок.*

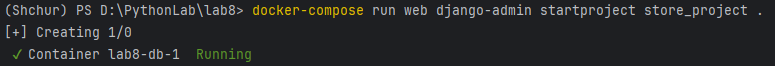
**docker-compose.yml:**

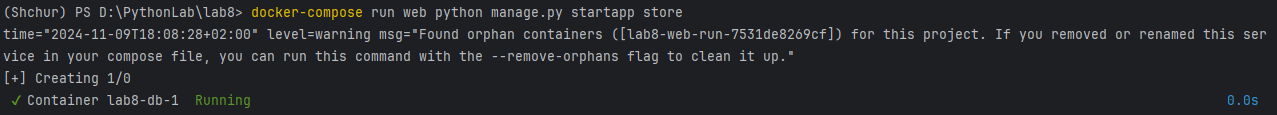
services:  
 db:  
 image: postgres:latest  
 environment:  
 POSTGRES\_USER: user  
 POSTGRES\_PASSWORD: password  
 POSTGRES\_DB: clothing\_store\_db  
 ports:  
 - "5432:5432"  
 volumes:  
 - db\_data:/var/lib/postgresql/data  
  
 web:  
 build: .  
 command: python manage.py runserver 0.0.0.0:8000  
 volumes:  
 - .:/app  
 ports:  
 - "8000:8000"  
 depends\_on:  
 - db  
  
volumes:  
 db\_data:

* Зробити прокидання портів та папок для підключення Django-застосунка до бази даних із контейнера, створеного в  *Лабораторній роботі  7*, відповідно до свого варіанту.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис





**models.py:**

from django.db import models  
  
class Warehouse(models.Model):  
 warehouse\_id = models.AutoField(primary\_key=True)  
 address = models.CharField(max\_length=255)  
 manager\_name = models.CharField(max\_length=100)  
 phone = models.CharField(max\_length=15)  
  
 def \_\_str\_\_(self):  
 return f"Warehouse {self.warehouse\_id} - {self.address}"  
  
class Product(models.Model):  
 PRODUCT\_TYPE\_CHOICES = [  
 ('Жіночий', 'Жіночий'),  
 ('Чоловічий', 'Чоловічий'),  
 ('Дитячий', 'Дитячий')  
 ]  
  
 product\_id = models.AutoField(primary\_key=True)  
 product\_type = models.CharField(max\_length=50, choices=PRODUCT\_TYPE\_CHOICES)  
 product\_name = models.CharField(max\_length=100)  
 manufacturer = models.CharField(max\_length=100)  
 warehouse = models.ForeignKey(Warehouse, on\_delete=models.CASCADE)  
 quantity\_on\_hand = models.IntegerField()  
 price = models.DecimalField(max\_digits=10, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"{self.product\_name} ({self.product\_type})"  
  
class Client(models.Model):  
 client\_id = models.AutoField(primary\_key=True)  
 client\_name = models.CharField(max\_length=100)  
 client\_address = models.CharField(max\_length=255)  
 phone = models.CharField(max\_length=15)  
 contact\_person = models.CharField(max\_length=100)  
  
 def \_\_str\_\_(self):  
 return self.client\_name  
  
class Sale(models.Model):  
 sale\_id = models.AutoField(primary\_key=True)  
 sale\_date = models.DateField()  
 client = models.ForeignKey(Client, on\_delete=models.CASCADE)  
 product = models.ForeignKey(Product, on\_delete=models.CASCADE)  
 quantity\_sold = models.IntegerField()  
 discount = models.DecimalField(max\_digits=5, decimal\_places=2)  
  
 def \_\_str\_\_(self):  
 return f"Sale {self.sale\_id} on {self.sale\_date}"

* Створити і налаштувати адміністративну панель Django для роботи з БД (додавання, редагування даних у всіх таблицях).

Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

**Виконуємо міграції:**

Зображення, що містить текст, програмне забезпечення, Мультимедійне програмне забезпечення, Комп’ютерна піктограма

Автоматично згенерований опис

**Створення адміністративного користувача:**

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

* Створити web сторінку (або сторінки) на яку вивести наступну інформацію:
* Назва проекту (заголовок 1 рівня)
* ПІБ студента, група (курсив, колір)
* Всі таблиці БД

**views.py:**

from django.shortcuts import render  
from .models import Medicine, Supplier, Supply  
  
def index(request):  
 medicines = Medicine.objects.all()  
 suppliers = Supplier.objects.all()  
 supplies = Supply.objects.all()  
 return render(request, 'pharmacy/index.html', {  
 'medicines': medicines,  
 'suppliers': suppliers,  
 'supplies': supplies,  
 })

**urls.py:**

from django.urls import path  
from .views import index  
  
urlpatterns = [  
 path('', index, name='index'),  
]

**index.html:**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Warehouse Management System</title>  
 <style>  
 h1 { text-align: center; }  
 h2 { text-align: center; }  
 em { color: blue; font-style: italic; }  
 table { width: 100%; border-collapse: collapse; }  
 th, td { padding: 8px; text-align: center; border: 1px solid black; }  
 th { font-weight: bold; }  
 </style>  
</head>  
<body>  
 <h1>Warehouse Management System</h1>  
 <h2>Shchur Veronika Mykolayivna, IPZ-21007b</h2>  
  
 <h2>Warehouses</h2>  
 <table>  
 <tr>  
 <th>Warehouse ID</th>  
 <th>Address</th>  
 <th>Manager Name</th>  
 <th>Phone</th>  
 </tr>  
 {% for warehouse in warehouses %}  
 <tr>  
 <td>{{ warehouse.warehouse\_id }}</td>  
 <td>{{ warehouse.address }}</td>  
 <td>{{ warehouse.manager\_name }}</td>  
 <td>{{ warehouse.phone }}</td>  
 </tr>  
 {% endfor %}  
 </table>  
  
 <h2>Products</h2>  
 <table>  
 <tr>  
 <th>Product ID</th>  
 <th>Product Type</th>  
 <th>Product Name</th>  
 <th>Manufacturer</th>  
 <th>Warehouse</th>  
 <th>Quantity on Hand</th>  
 <th>Price</th>  
 </tr>  
 {% for product in products %}  
 <tr>  
 <td>{{ product.product\_id }}</td>  
 <td>{{ product.product\_type }}</td>  
 <td>{{ product.product\_name }}</td>  
 <td>{{ product.manufacturer }}</td>  
 <td>{{ product.warehouse.address }}</td>  
 <td>{{ product.quantity\_on\_hand }}</td>  
 <td>{{ product.price }}</td>  
 </tr>  
 {% endfor %}  
 </table>  
  
 <h2>Clients</h2>  
 <table>  
 <tr>  
 <th>Client ID</th>  
 <th>Client Name</th>  
 <th>Client Address</th>  
 <th>Phone</th>  
 <th>Contact Person</th>  
 </tr>  
 {% for client in clients %}  
 <tr>  
 <td>{{ client.client\_id }}</td>  
 <td>{{ client.client\_name }}</td>  
 <td>{{ client.client\_address }}</td>  
 <td>{{ client.phone }}</td>  
 <td>{{ client.contact\_person }}</td>  
 </tr>  
 {% endfor %}  
 </table>  
  
 <h2>Sales</h2>  
 <table>  
 <tr>  
 <th>Sale ID</th>  
 <th>Sale Date</th>  
 <th>Client</th>  
 <th>Product</th>  
 <th>Quantity Sold</th>  
 <th>Discount</th>  
 </tr>  
 {% for sale in sales %}  
 <tr>  
 <td>{{ sale.sale\_id }}</td>  
 <td>{{ sale.sale\_date }}</td>  
 <td>{{ sale.client.client\_name }}</td>  
 <td>{{ sale.product.product\_name }}</td>  
 <td>{{ sale.quantity\_sold }}</td>  
 <td>{{ sale.discount }}</td>  
 </tr>  
 {% endfor %}  
 </table>  
</body>  
</html>

**Результат:**

Зображення, що містить текст, квитанція, число, Паралель

Автоматично згенерований опис

***2.****Створити файл requirments.txt, в якому зберігаються всі встановлені модулі та пакети.*



Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

***3.****Завантажити проект на GitHub, попередньо додавши до файлу .gitignore всі технічні папки та файли.*