



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №3
З дисципліни «Основи Back-end технологій»

Виконав:

студент групи ІП-13
Карамян В.С.

Перевірив:

викладач
Зубко Р.А.

Київ 2024

Лабораторна робота №3

Тема:

Python&PostgresSQL. Створення сторінки статистики сайту. Пошук інформації по сайту.

Завдання:

1. Зробити виведення на сторінку статистики наступних даних web-сайту:
 - Скільки всього було зроблено записів у табл.1;
 - Скільки було зроблено записів у табл.2;
 - Скільки записів було зроблено за останній місяць у табл.1 та табл.2;
 - Який запис був зроблений останнім (у табл.1);
 - Який запис у табл.1 має найбільшу кількість пов'язаних записів у табл.2.
2. Виконати пошук інформації в БД за ключовим словом, шаблоном та у заданому діапазоні

Варіант 2

Спроектувати базу даних про автомобілі: номер, рік випуску, марка, колір, стан, прізвище власника, адреса.

Хід роботи

1. Зробити виведення на сторінку статистики наступних даних web-сайту:

Додамо в наші таблиці поле з датою створення:

Змінимо моделі:

```
class Car(models.Model):
    """Car Model"""
    license_plate = models.CharField(max_length=10, unique=True)
    brand = models.CharField(max_length=255)
    year = models.IntegerField()
    color = models.CharField(max_length=255)
    condition = models.CharField(max_length=255)
    owner = models.ForeignKey(Owner, on_delete=models.CASCADE)
    date_created = models.DateTimeField(auto_now_add=True)
```

Проведемо міграції:

```
(venv) PS D:\ProjectsPy\kpi-projects\backend-labs\car-app-api\app> python .\manage.py makemigrations
Migrations for 'car':
  car\migrations\0002_car_date_created_owner_date_created.py
    - Add field date_created to car
    - Add field date_created to owner
(venv) PS D:\ProjectsPy\kpi-projects\backend-labs\car-app-api\app> migrate
```

Додаємо новий endpoint /statistics:

```
path('statistics/', statistics, name='statistics')
```

Та представлення для обробки запиту, де ми знаходимо загальну кількість записів у таблицях:

```
2 usages new *
def statistics(request):
    # Total records in Owner table
    owner_count = Owner.objects.count()

    # Total records in Car table
    car_count = Car.objects.count()
```

та іншу статистику:

```
# Total records in Owner table for the last month
last_month = timezone.now() - timezone.timedelta(days=30)
owner_count_last_month = Owner.objects.filter(date_created__gte=last_month).count()

# Total records in Owner table for the last month
car_count_last_month = Car.objects.filter(date_created__gte=last_month).count()

# The last record in Owner table
last_owner_record = Owner.objects.latest('date_created')

# The record in Owner table with the largest number of related records in Car table
owner_with_most_cars = Owner.objects.annotate(num_cars=Count('car')).order_by('-num_cars').first()
```

Передаємо ці дані у шаблон:

```
context = {
    'owner_count': owner_count,
    'car_count': car_count,
    'owner_count_last_month': owner_count_last_month,
    'car_count_last_month': car_count_last_month,
    'last_owner_record': last_owner_record,
    'owner_with_most_cars': owner_with_most_cars,
}

return render(request, template_name='statistics.html', context)
```

У вигляді таблиці виводимо дані:

```
<table>
  <tbody>
    <tr>
      <td>Total records in Owner table:</td>
      <td>{{ owner_count }}</td>
    </tr>
    <tr>
      <td>Total records in Car table:</td>
      <td>{{ car_count }}</td>
    </tr>
    <tr>
      <td>Records created in the last month in Owner table:</td>
      <td>{{ owner_count_last_month }}</td>
    </tr>
  </tbody>
</table>
```

Демонстрація роботи

Таблиця з машинами наразі має такий вигляд:

Car List

Sort by: ID

License Plate	Brand	Year	Color	Condition	Owner	Edit	Delete
AB1784HT	Skoda	2004	Black	Used	Vartan Karamian	Edit	Delete
KA2804EM	Volkswagen	2002	Gray	Used	Vartan Karamian	Edit	Delete
AD1244OP	Ferarri	2021	Red	New	Michael White	Edit	Delete
AD9881DF	Audi	2010	White	Used	Vartan Karamian	Edit	Delete
AD1244OR	Toyota	2024	Black	New	Michael White	Edit	Delete
7J2398	Lamborghini	2024	Yellow	New	John Smith	Edit	Delete

Add new car

Переходимо на сторінку зі статистикою:

Statistics

Total records in Owner table:	3
Total records in Car table:	6
Records created in the last month in Owner table:	3
Records created in the last month in Car table:	6
The last record in Owner table:	John Smith
Owner with the most associated records in Car table:	Vartan Karamian

2. Виконати пошук інформації в БД за ключовим словом, шаблоном та у заданому діапазоні

Додамо форму для пошуку за ключовим словом та діапазоном:

```
3 usages new *
class CarSearchForm(forms.Form):
    keyword = forms.CharField(label='Keyword', max_length=100, required=False)
    min_year = forms.IntegerField(required=False, label='Min Year')
    max_year = forms.IntegerField(required=False, label='Max Year')
```

Змінимо представлення для перегляду списку машин:

```
19 def car_table(request):
20     """
21     View for displaying the car table.
22     """
23     if request.method == 'GET':
24         form = CarSearchForm(request.GET)
25         if form.is_valid():
26             keyword = form.cleaned_data.get('keyword')
27             min_year = form.cleaned_data.get('min_year')
28             max_year = form.cleaned_data.get('max_year')
29             sort_by = request.GET.get('sort_by', 'id')
30
31             cars = Car.objects.all().order_by(sort_by)
32
33             if keyword:
34                 cars = cars.filter(brand__icontains=keyword)
35             if min_year:
36                 cars = cars.filter(year__gte=min_year)
37             if max_year:
38                 cars = cars.filter(year__lte=max_year)
39
40         else:
41             form = CarSearchForm()
42             cars = Car.objects.all()
43
44     <> return render(request, template_name='car_table.html', context={'cars': cars, 'form': form})
```

Додамо в HTML шаблон поля для пошуку:

```

<form method="get">
  {% csrf_token %}
  <div class="search-elements">
    <div class="search-title">
      <h2>Search</h2>
    </div>
    <div class="search-input">
      <div class="brand-search">
        <p>Brand</p>
        {{ form.keyword }}
      </div>
      <div class="min-year">
        <p>Year from</p>
        {{ form.min_year }}
      </div>
      <div class="max-year">
        <p>Year to</p>
        {{ form.max_year }}
      </div>
      <button type="submit" class="search-button">Search</button>
    </div>
  </div>
</form>

```

Демонстрація роботи програми

Відкриємо головну сторінку:

Car List

Sort by: ID

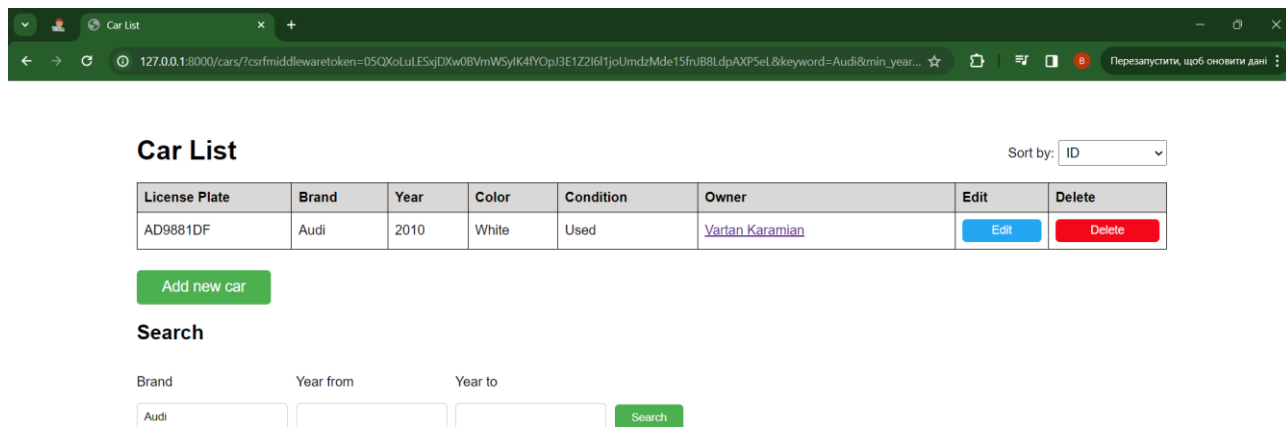
License Plate	Brand	Year	Color	Condition	Owner	Edit	Delete
AB1784HT	Skoda	2004	Black	Used	Vartan Karamian	Edit	Delete
KA2804EM	Volkswagen	2002	Gray	Used	Vartan Karamian	Edit	Delete
AD1244OP	Ferarri	2021	Red	New	Michael White	Edit	Delete
AD9881DF	Audi	2010	White	Used	Vartan Karamian	Edit	Delete
AD1244OR	Toyota	2024	Black	New	Michael White	Edit	Delete
7J2398	Lamborghini	2024	Yellow	New	John Smith	Edit	Delete
KA2804DF	BMW	1987	Black	Restored	John Smith	Edit	Delete

[Add new car](#)

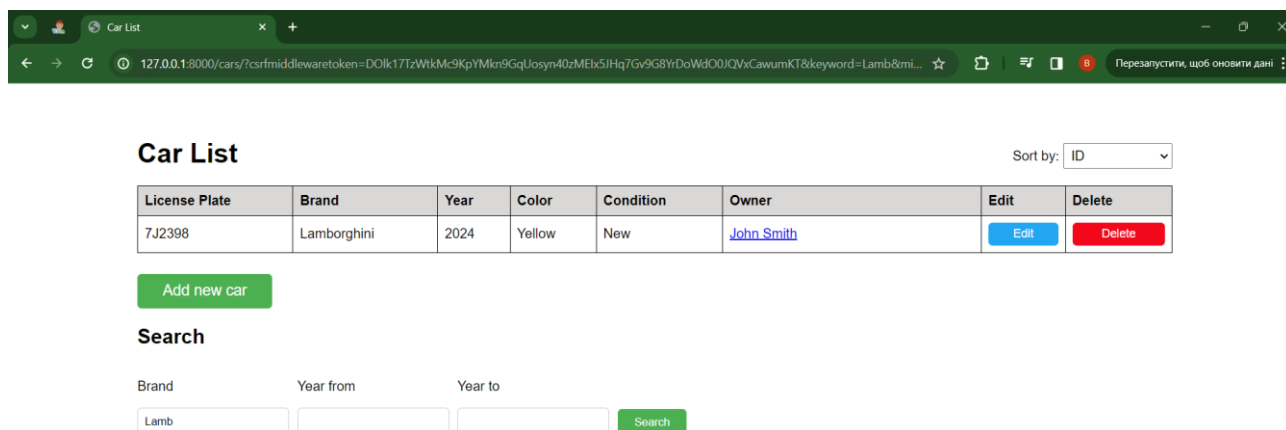
Search

Brand Year from Year to [Search](#)

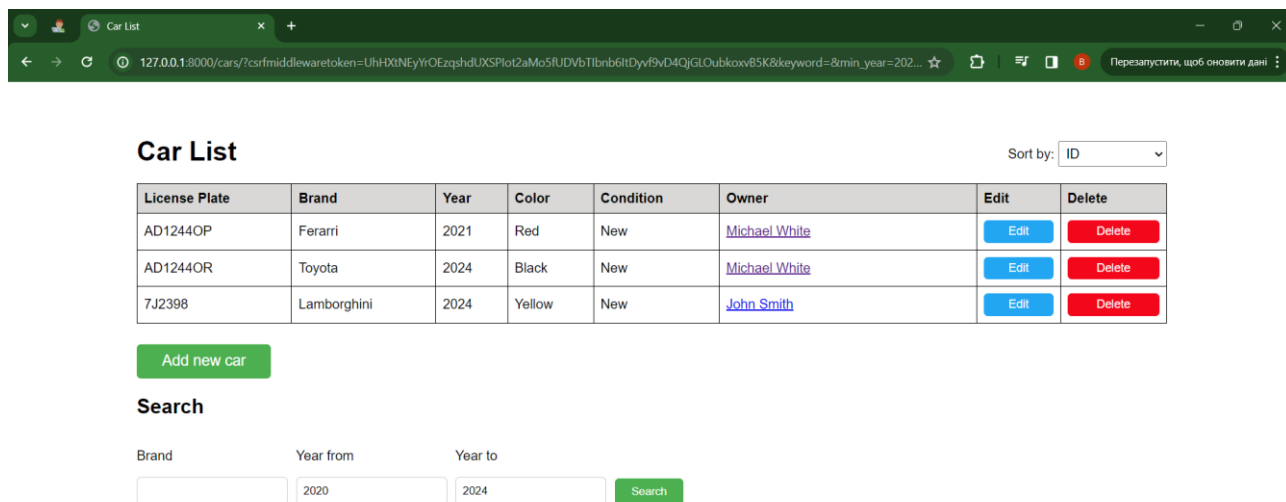
Здійснимо пошук за ключовим словом:



Пошук за шаблоном:



Пошук в діапазоні років:



Код програми:

<https://github.com/Vartan14/car-app-api/>

Висновок

Під час виконання лабораторної роботи я вивчив реалізацію створення сторінки статистики для веб-сайту за допомогою Python та PostgreSQL. Для цього я використовував Django, який дозволяє легко працювати з базами даних та реалізовувати веб-сторінки.

Було виконано виведення на сторінку статистики різних даних з бази даних, таких як загальна кількість записів у таблицях, кількість записів за останній місяць та інші. Я отримав досвід написання функціоналу пошуку інформації в базі даних за ключовим словом, шаблоном та у заданому діапазоні.

Під час виконання цих завдань я отримав практичні навички з Django ORM для взаємодії з базою даних PostgreSQL, а також збагатив свої знання щодо роботи з шаблонами Django та реалізації різноманітного функціоналу на веб-сторінках.