partAdvisor

Caner Ayrancı

part Advisor

Araç sahiplerine araçları ve kullanım alışkanlıklarına uygun şekilde lastik ve ileride dahasını tavsiye etmek amacıyla başlanmış bir projedir.
Projenin kesinlikle bir ürün satış çabası olmamakla beraber tamamen kullanıcıların kendilerine en uygun marka ve modelleri önceden keşfetmesini hedeflemektedir.

partAdvisor nedir?

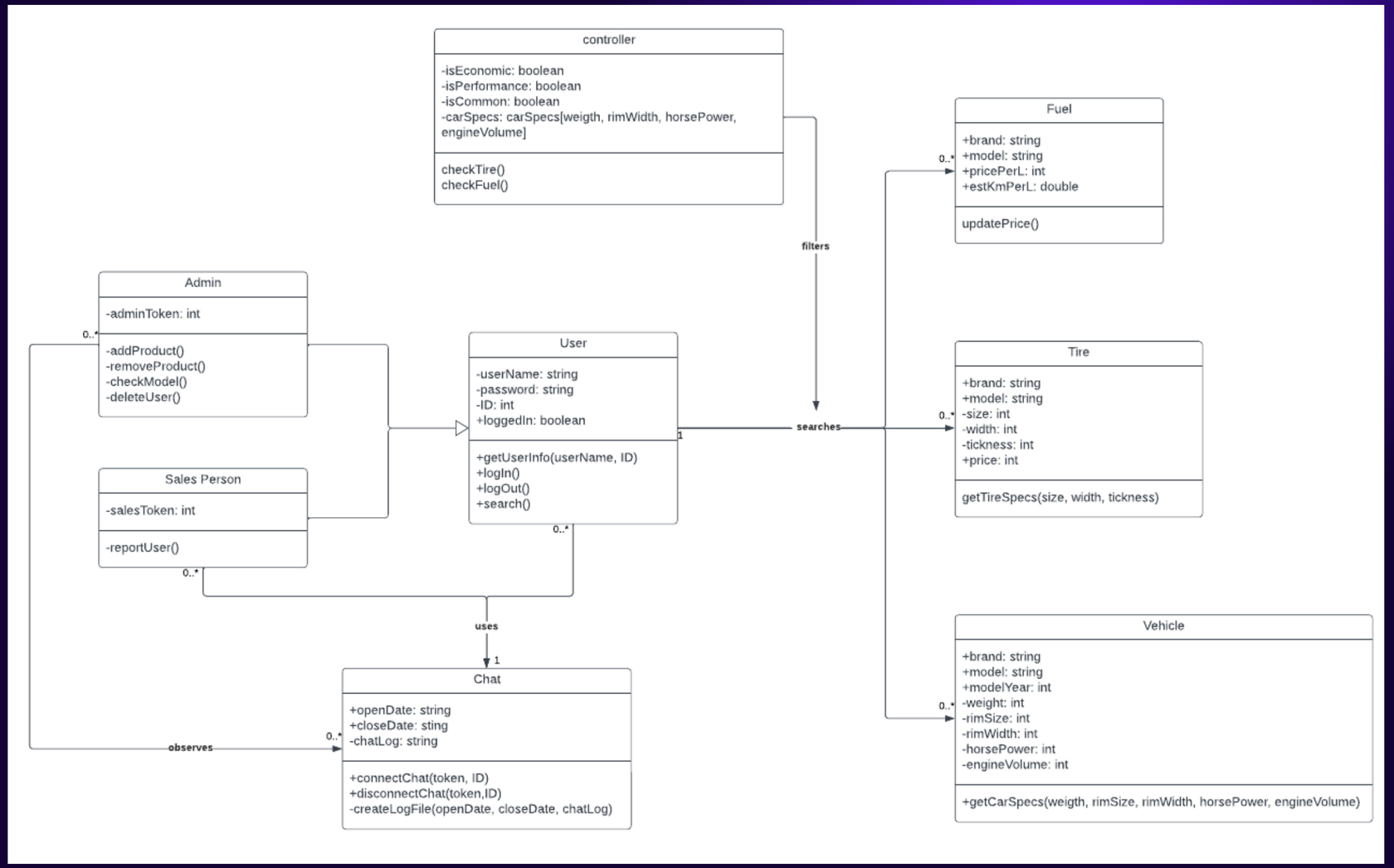
•Desteklenen markalardaki desteklenen modeller arasında kullanıcının araç marka ve modeline en uygun, tahmini genel yükü ele alınarak, en ekonomik, günlük kullanım ya da performansa dayalı lastik tavsiyesi yapabilmektir.

partAdvisor

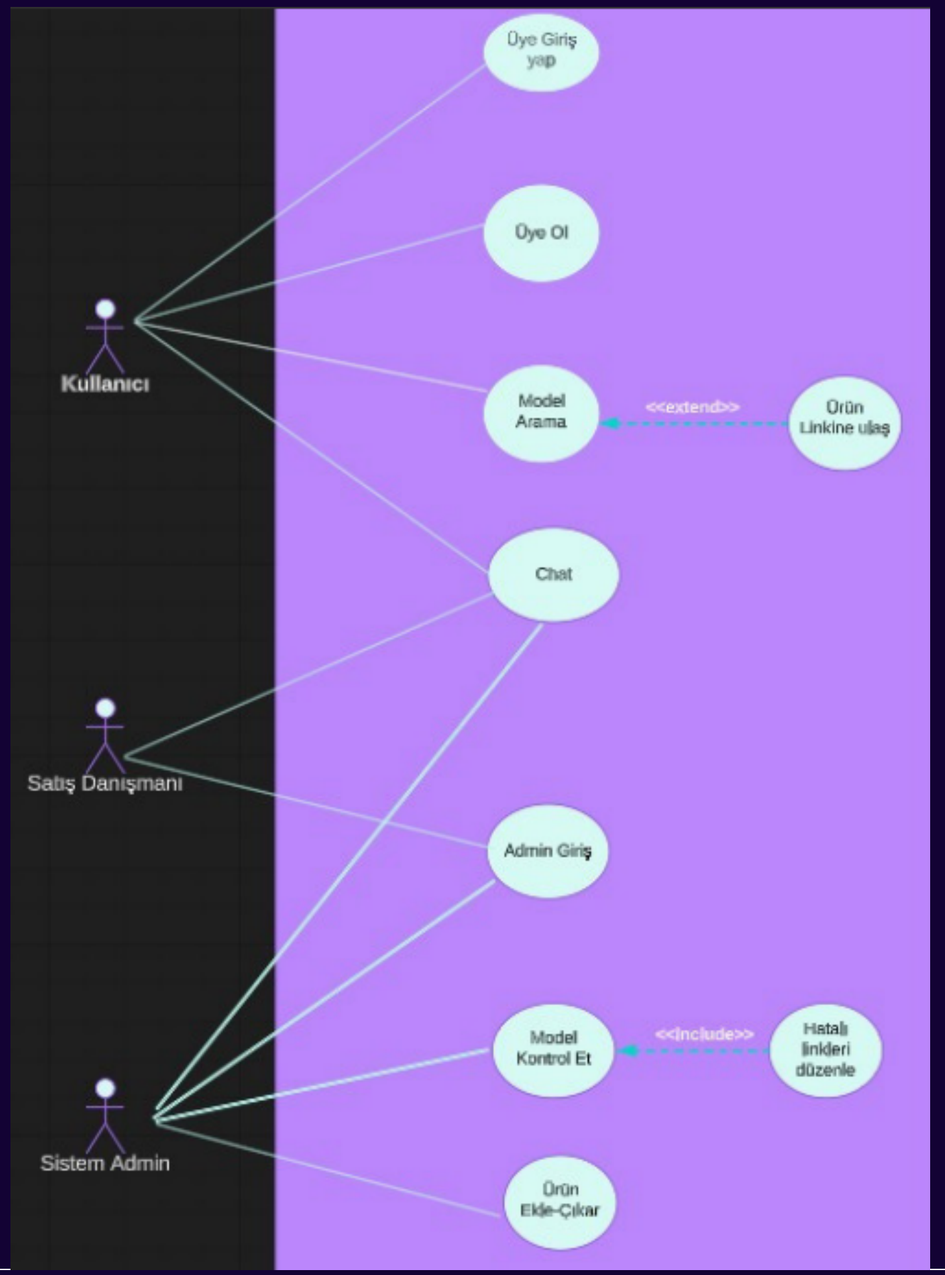
Diyagramlar

- Class Diyagramı
- Use - Case Diyagramı
- Sequence Diyagramı
- ER Diyagramı

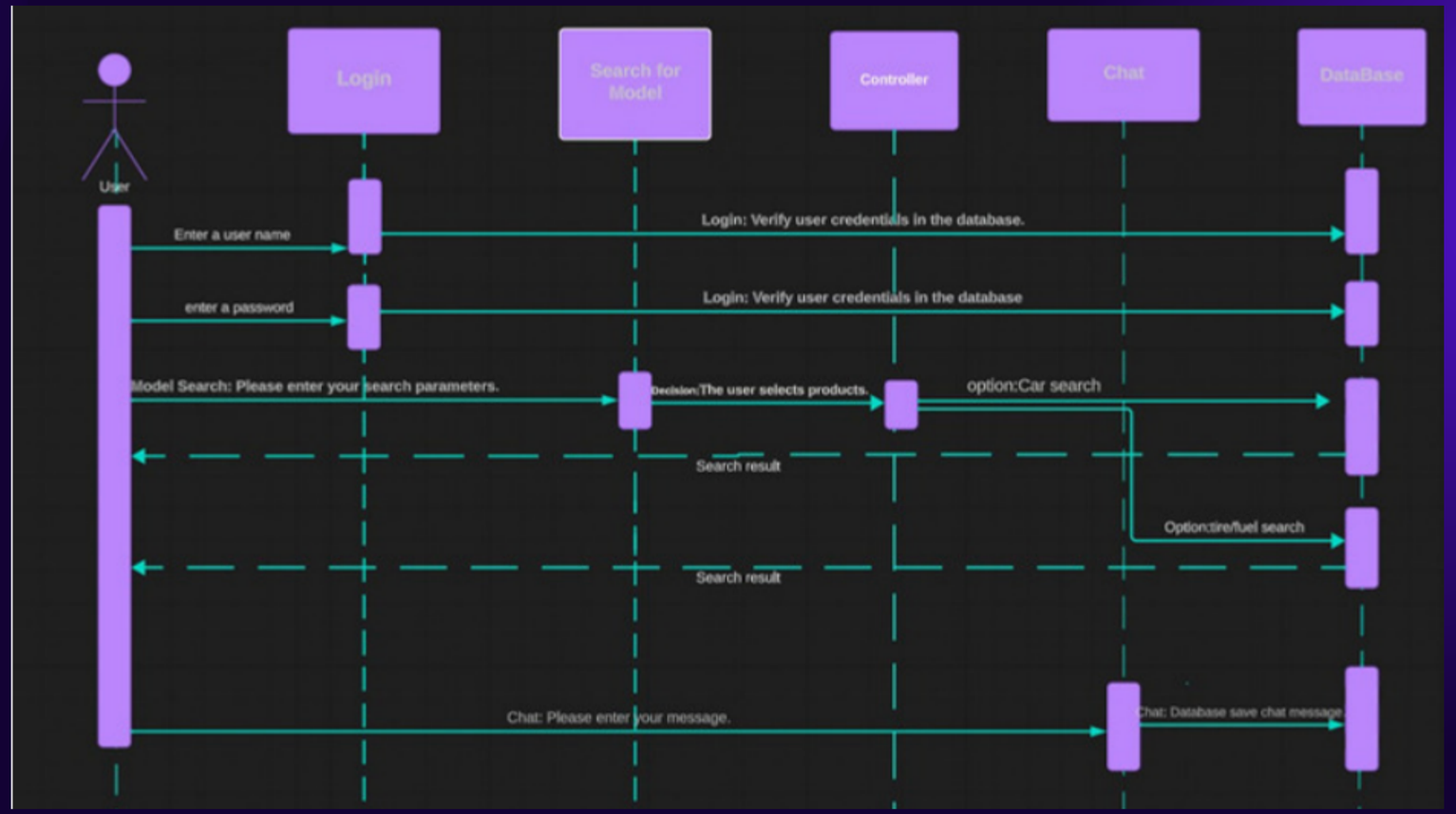
Class Diyagramı



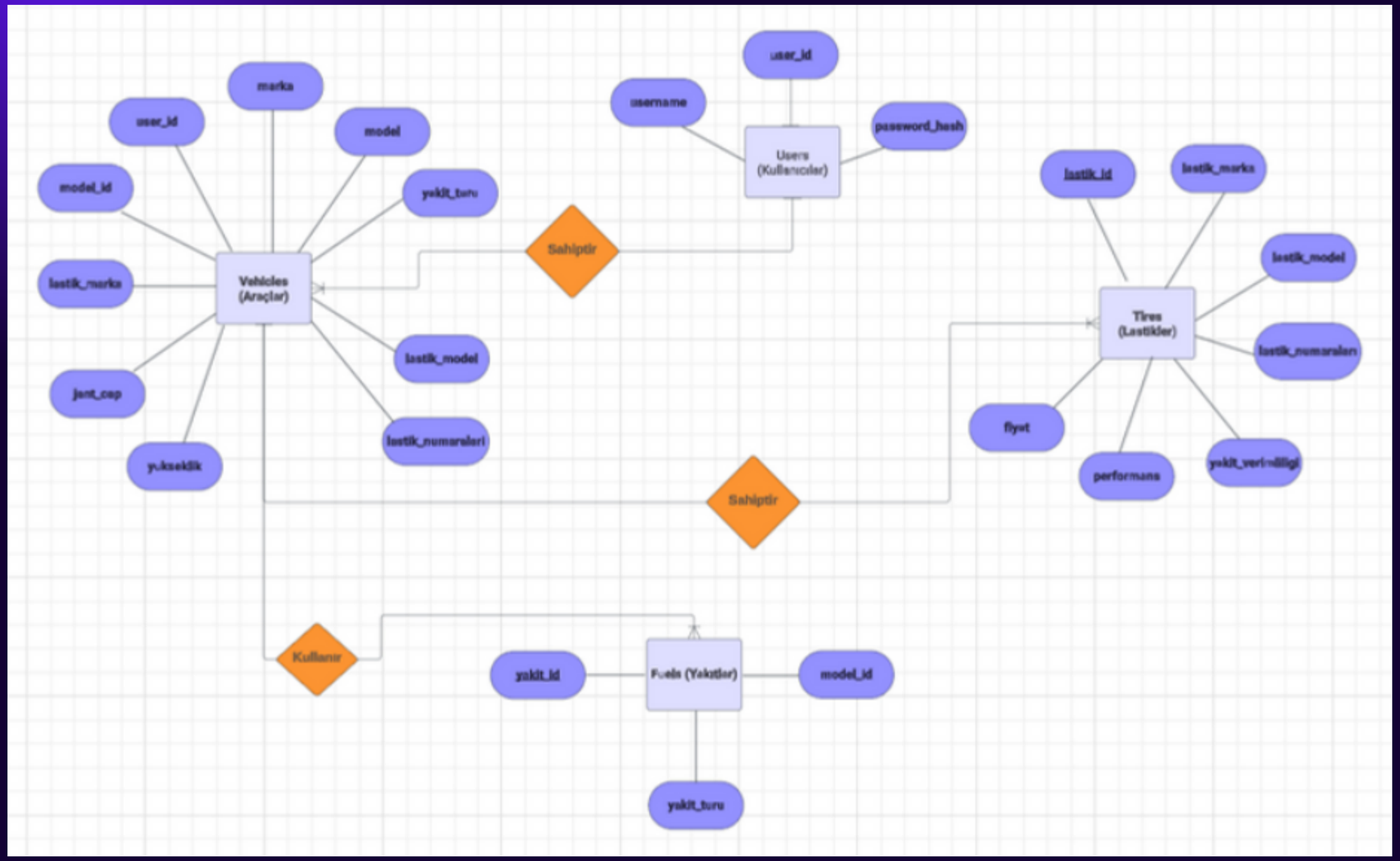
Use - Case Diyagramı



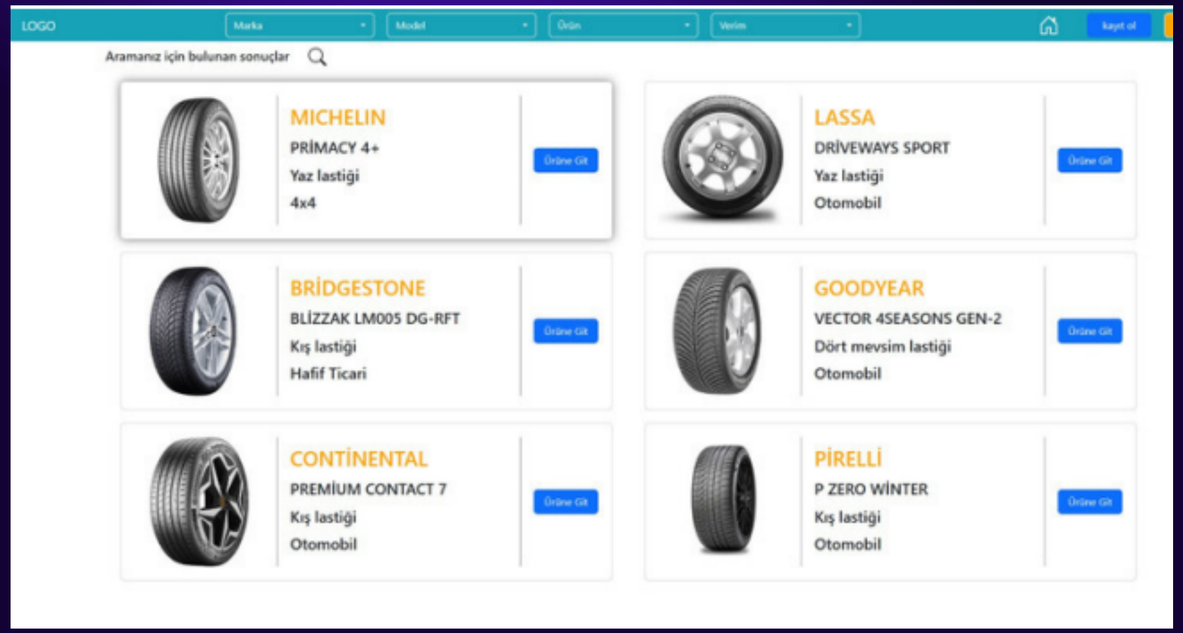
Sequence Diyagramı

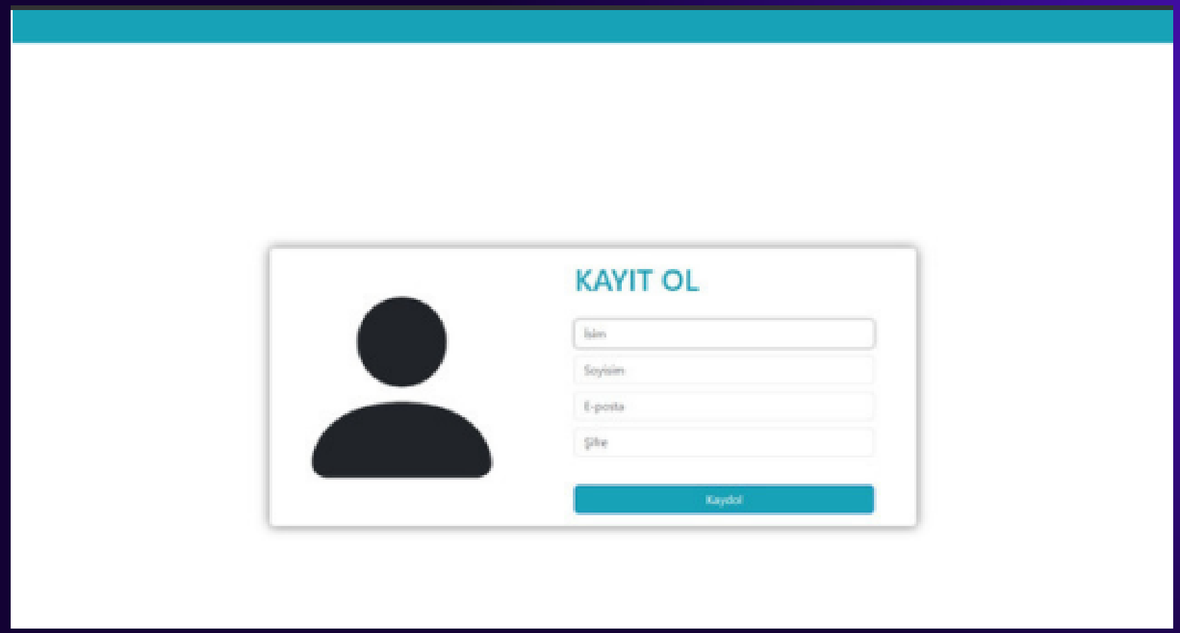


ER Diyagramı



Demo UI


LOGO
Marka
Model
Ürün
Verim
kayıt ol
Aramanız için bulunan sonuçlar
MICHELIN
PRİMACY 4+
Yaz lastiği
4x4
Ürüne Git
LASSA
DRİVEWAYS SPORT
Yaz lastiği
Otomobil
Ürüne Git
BRİDGESTONE
BLİZZAK LM005 DG-RFT
Kış lastiği
Hafif Ticari
Ürüne Git
GOODYEAR
VECTOR 4SEASONS GEN-2
Dört mevsim lastiği
Otomobil
Ürüne Git
CONTİNENTAL
PREMİUM CONTACT 7
Kış lastiği
Otomobil
Ürüne Git
PİRELLİ
P ZERO WİNTER
Kış lastiği
Otomobil
Ürüne Git


KAYIT OL
İsim
Soyisim
E-posta
Şifre
Kaydol

Backend Service

```
PS D:\Solast\DEV\partAdvisorAPI> . .\.venv\Scripts\activate
(.venv) PS D:\Solast\DEV\partAdvisorAPI> python.exe .\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 25, 2023 - 00:20:09
Django version 5.0, using settings 'backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Backend tamamen Python dili ve Django framework'ü üzerinde yazılmıştır. Virtual Environment (venv) yapısı ile local içerisindeki diğer projeler ile gereklilik çakışmaları önlenmektedir

Veriler Django üzerinde oluşturulan modeller ile framwork'ün oluşturacağı sqlite3 veritabanında tutuluyor

```python
from django.db import models

class Tire(models.Model):

    id = models.IntegerField(primary_key=True)
    model_id = models.IntegerField()
    brand = models.CharField(max_length = 200)
    model = models.CharField(max_length = 400)
    tireSpecs = models.CharField(max_length= 10)
    mevsim = models.CharField(max_length= 100)
    fuelNote = models.CharField(max_length= 1)
    performance = models.CharField(max_length= 25)
    price = models.IntegerField()

    def __str__(self):
        return self.brand + ' ' + self.model
```

```python
class Vehicle(models.Model):

    model_id = models.IntegerField(primary_key=True)
    user_id = models.IntegerField()
    brand = models.CharField(max_length= 200)
    model = models.CharField(max_length= 400)
    fuelType = models.CharField(max_length= 100)
    tireSpecs = models.CharField(max_length=10)
    tireWidth = models.IntegerField()
    tireHeight = models.IntegerField()
    rimSize = models.IntegerField()

    def __str__(self):
        return self.brand + ' ' + self.model
```

```python
class User(models.Model):

    user_id = models.IntegerField(primary_key=True)
    username = models.CharField(max_length= 20)
    email = models.CharField(max_length= 100)
    password = models.CharField(max_length= 20)
    model_id = models.IntegerField(null = True, blank = True)
```

URLS

```python
from django.contrib import admin
from django.urls import path
from backend import views
from rest_framework.urlpatterns import format_suffix_patterns

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.homePage),
    path('api/', views.apiPage),
    path('api/tires/', views.tireList),
    path('api/vehicles/', views.vehicleList),
    path('api/tires/<int:id>/', views.tireDetail),
    path('api/vehicles/<int:model_id>/', views.vehicleDetail),
]

urlpatterns = format_suffix_patterns(urlpatterns)
```

VIEWS

```python
@api_view(['GET', 'POST'])
def tireList(request, format=None):

    if request.method == 'GET':
        tires = Tire.objects.all()
        serializer = TireSerializer(tires, many=True)
        return Response(serializer.data)

    if request.method == 'POST':
        serializer = TireSerializer(data = request.data, many = True)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
```

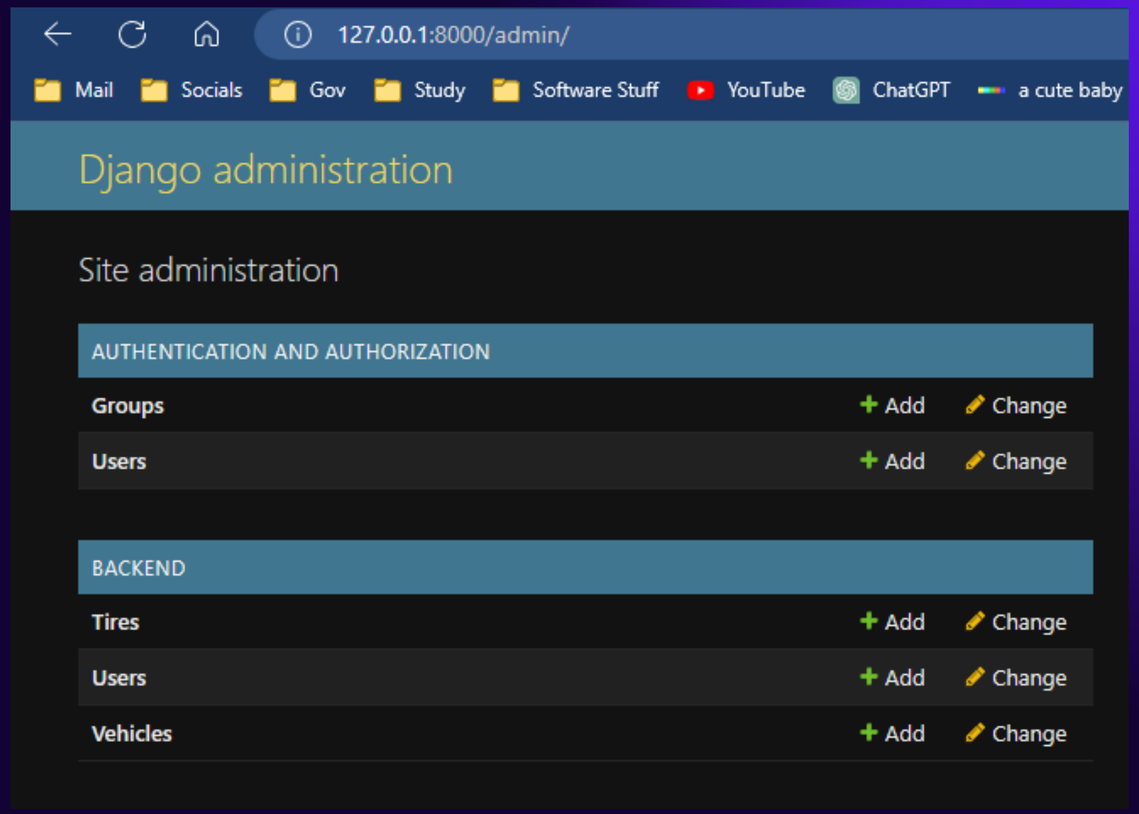
```python
@api_view(['GET', 'PUT', 'DELETE'])
def tireDetail(request, id, format=None):

    try:
        tires = Tire.objects.get(pk=id)
    except Tire.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    if request.method == 'GET':
        serializer = TireSerializer(tires)
        return Response(serializer.data)

    elif request.method == 'PUT':
        serializer = TireSerializer(tires, data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    elif request.method == 'DELETE':
        tires.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)
```

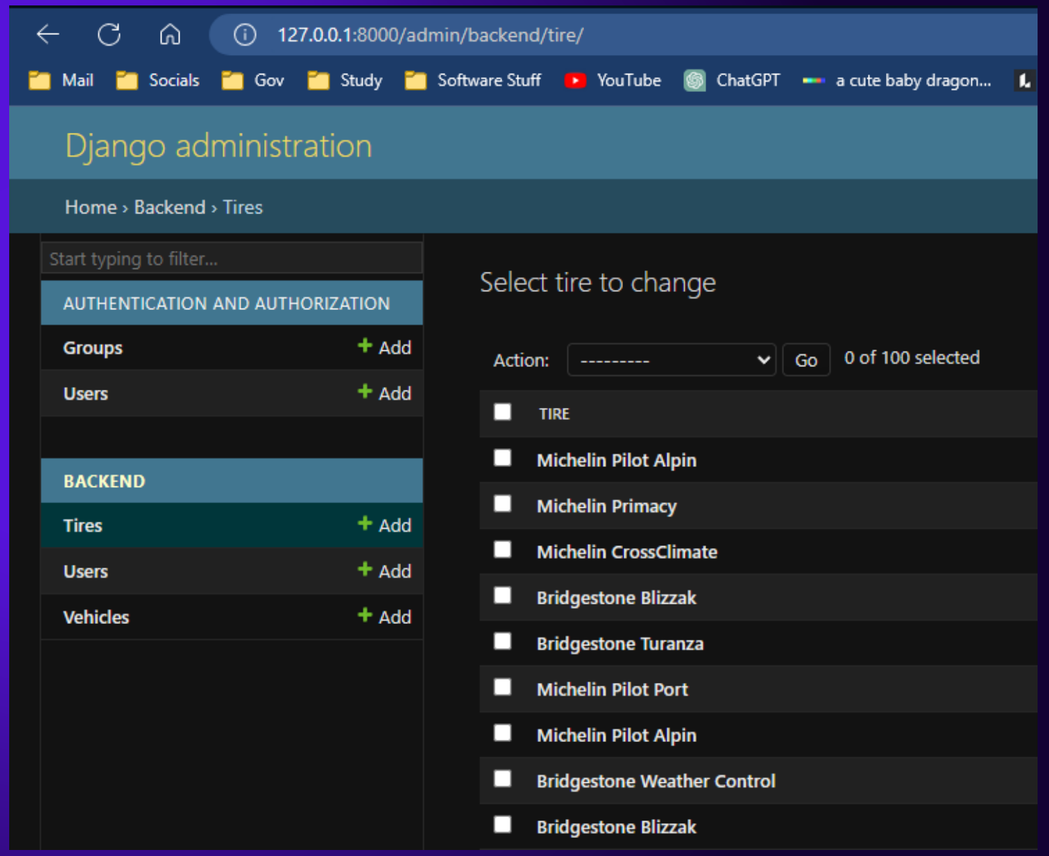
SERIALIZER

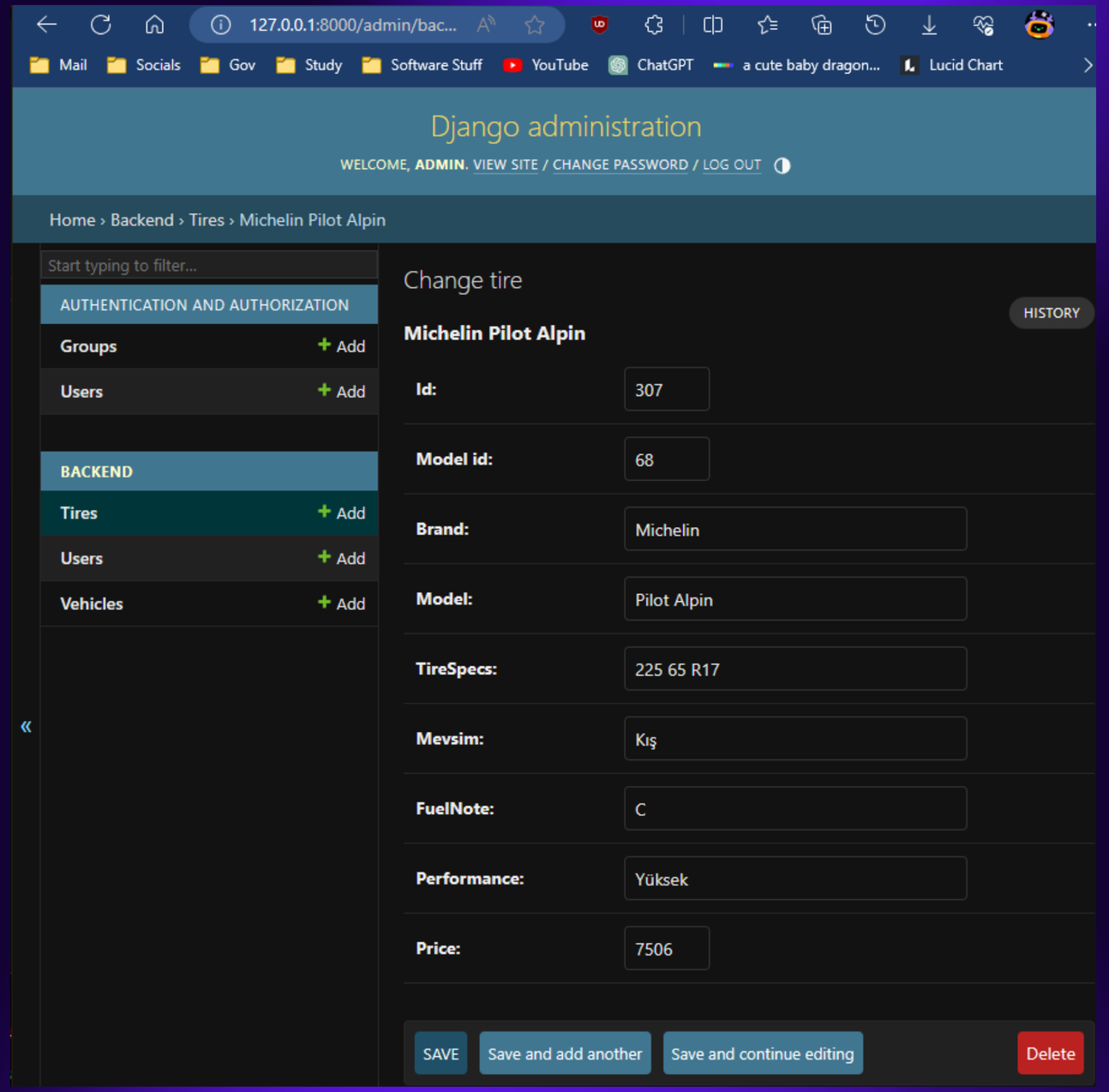
```python
from rest_framework import serializers
from .models import Tire, Vehicle

class TireSerializer(serializers.ModelSerializer):
    class Meta:
        model = Tire
        fields = ['id', 'model_id', 'brand', 'model', 'tireSpecs', 'mevsim', 'fuelNote', 'performance', 'price']

class VehicleSerializer(serializers.ModelSerializer):
    class Meta:
        model = Vehicle
        fields = ['model_id', 'user_id', 'brand', 'model', 'fuelType', 'tireSpecs', 'tireWidth','tireHeight', 'rimSize']
```

/admin

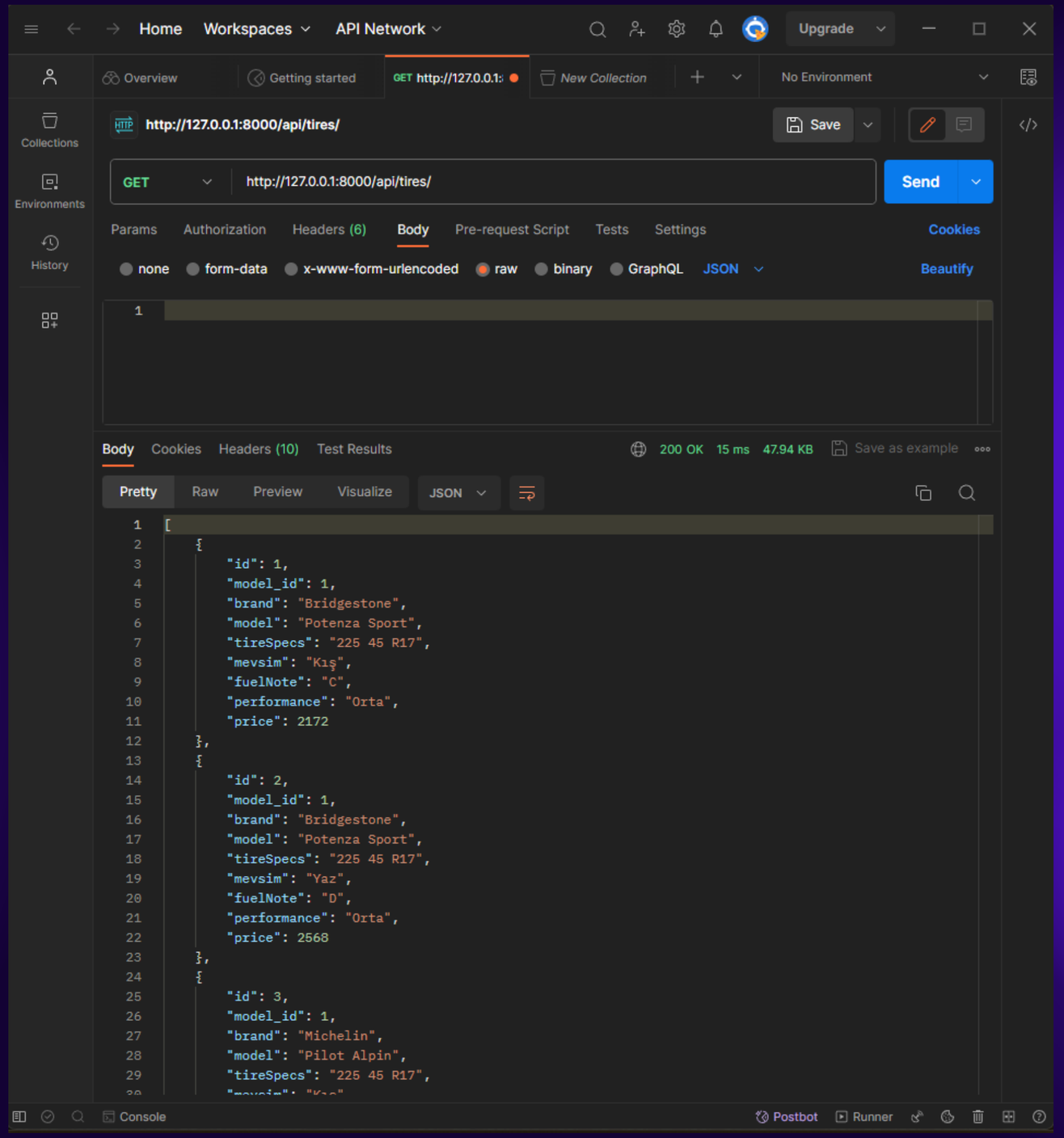
```python
from django.contrib import admin
from .models import Tire, Vehicle, User

admin.site.register(User)
admin.site.register(Tire)
admin.site.register(Vehicle)
```

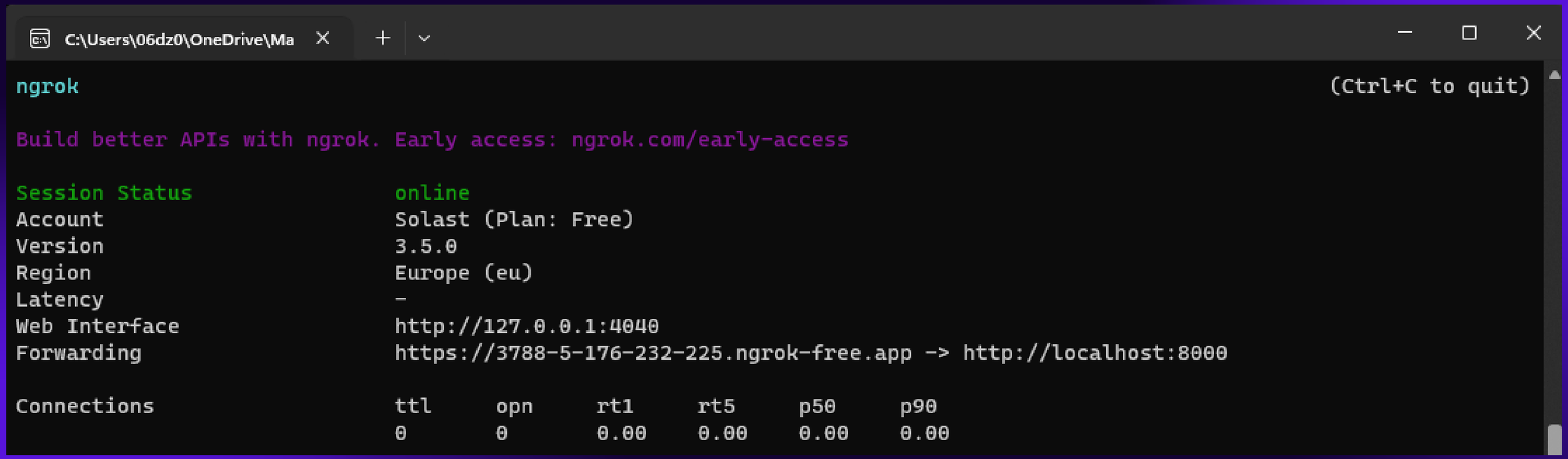


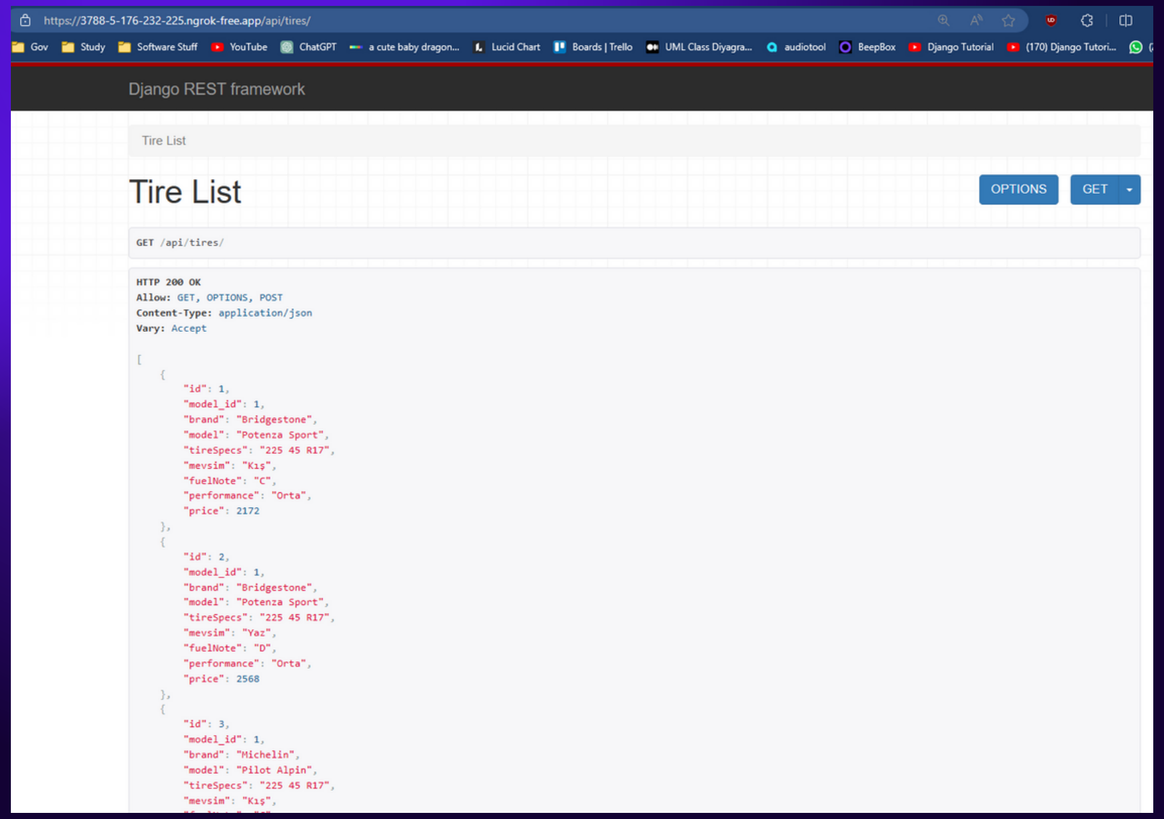

127.0.0.1:8000/admin/backend/tire/
Mail
Socials
Gov
Study
Software Stuff
YouTube
ChatGPT
a cute baby dragon...
Django administration
Home › Backend › Tires
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
BACKEND
Tires + Add
Users + Add
Vehicles + Add
Select tire to change
Action: --------- Go 0 of 100 selected
TIRE
Michelin Pilot Alpin
Michelin Primacy
Michelin CrossClimate
Bridgestone Blizzak
Bridgestone Turanza
Michelin Pilot Port
Michelin Pilot Alpin
Bridgestone Weather Control
Bridgestone Blizzak

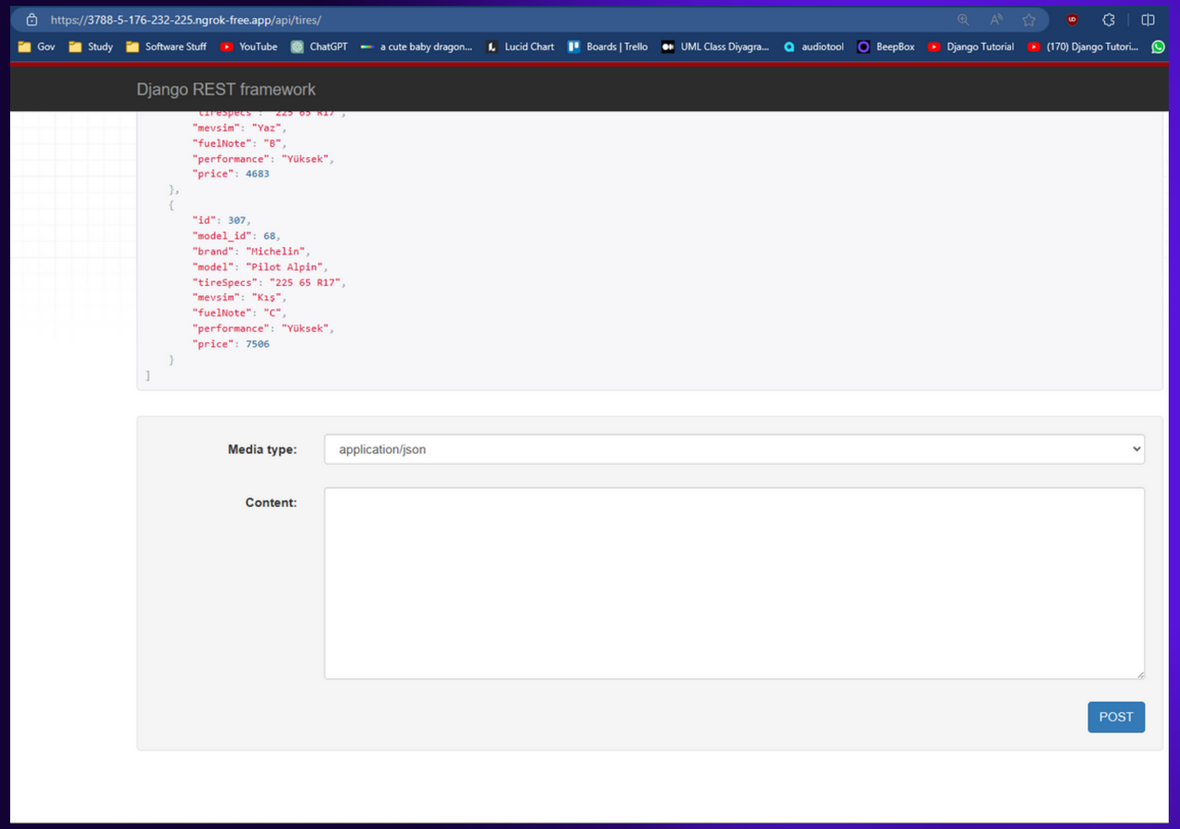

127.0.0.1:8000/admin/bac...
Mail
Socials
Gov
Study
Software Stuff
YouTube
ChatGPT
a cute baby dragon...
Lucid Chart
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Backend › Tires › Michelin Pilot Alpin
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
BACKEND
Tires + Add
Users + Add
Vehicles + Add
Change tire
HISTORY
Michelin Pilot Alpin
Id: 307
Model id: 68
Brand: Michelin
Model: Pilot Alpin
TireSpecs: 225 65 R17
Mevsim: Kış
FuelNote: C
Performance: Yüksek
Price: 7506
SAVE
Save and add another
Save and continue editing
Delete

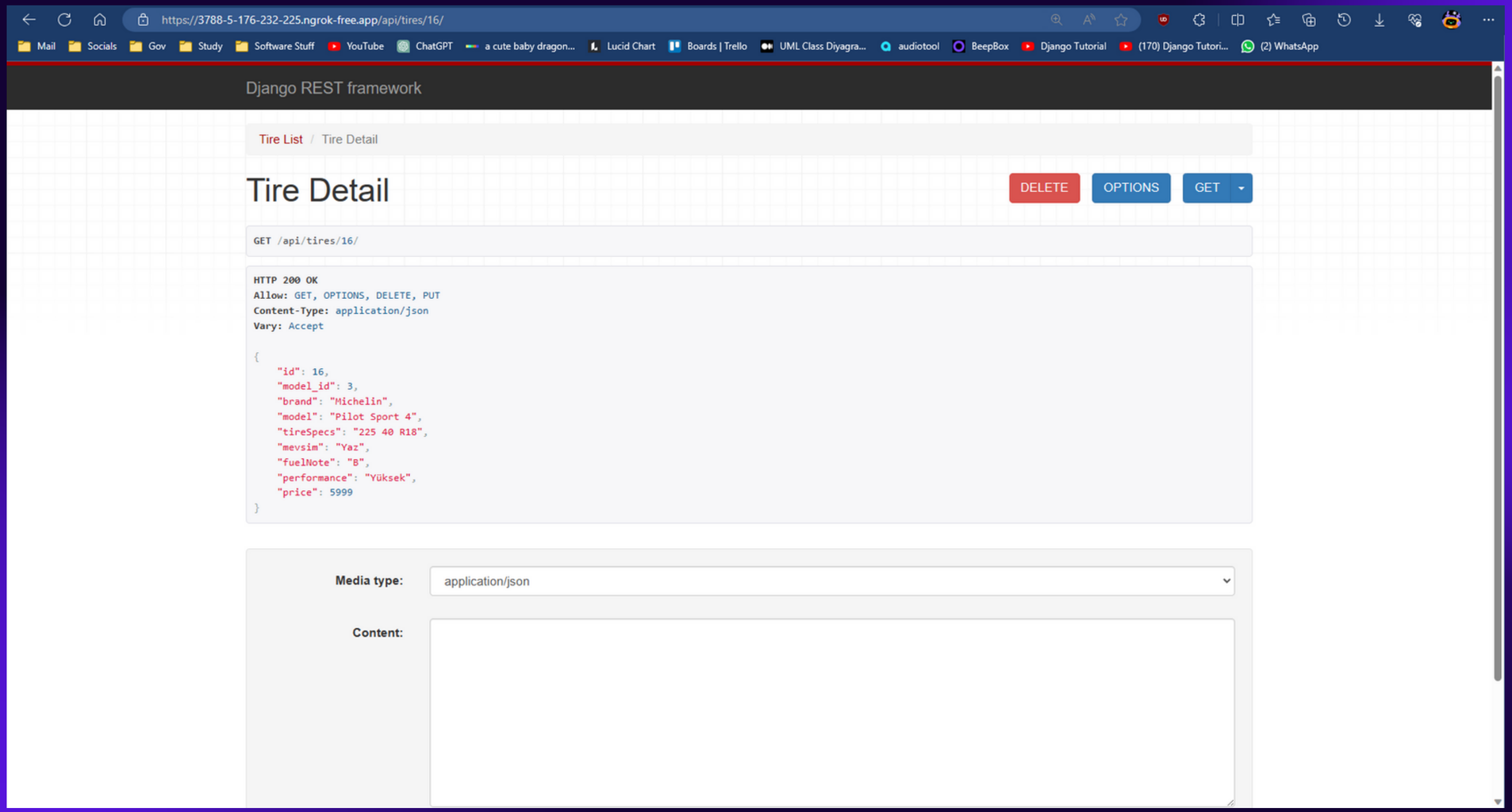
```
[25/Dec/2023 00:35:01] "GET /admin/ HTTP/1.1" 200 10328
[25/Dec/2023 00:35:01] "GET /static/admin/css/base.css HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/css/dark_mode.css HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/css/dashboard.css HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/css/responsive.css HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/js/theme.js HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 304 0
[25/Dec/2023 00:35:01] "GET /static/admin/img/icon-deletelink.svg HTTP/1.1" 304 0
[25/Dec/2023 00:35:59] "GET /admin/backend/tire/ HTTP/1.1" 200 38694
[25/Dec/2023 00:36:00] "GET /static/admin/css/changelists.css HTTP/1.1" 304 0
[25/Dec/2023 00:36:00] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 304 0
[25/Dec/2023 00:36:00] "GET /static/admin/js/jquery.init.js HTTP/1.1" 304 0
[25/Dec/2023 00:36:00] "GET /static/admin/js/core.js HTTP/1.1" 304 0
[25/Dec/2023 00:36:00] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 304 0
[25/Dec/2023 00:36:00] "GET /static/admin/js/actions.js HTTP/1.1" 304 0
```

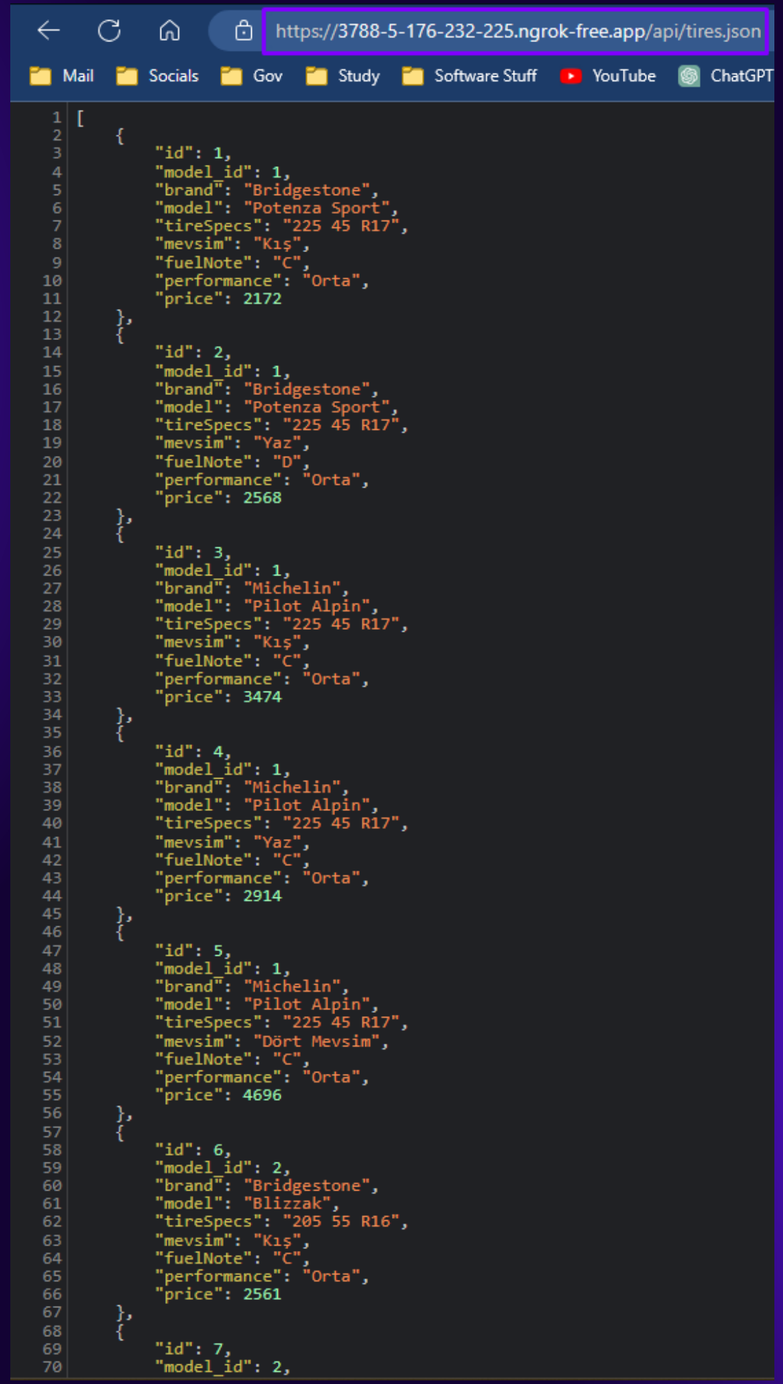

Home
Workspaces
API Network
Upgrade
Overview
Getting started
GET http://127.0.0.1:
New Collection
No Environment
Collections
Environments
History
http://127.0.0.1:8000/api/tires/
Save
GET
http://127.0.0.1:8000/api/tires/
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
Body
Cookies
Headers (10)
Test Results
200 OK
15 ms
47.94 KB
Save as example
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"model_id": 1,
"brand": "Bridgestone",
"model": "Potenza Sport",
"tireSpecs": "225 45 R17",
"mevsim": "Kış",
"fuelNote": "C",
"performance": "Orta",
"price": 2172
},
{
"id": 2,
"model_id": 1,
"brand": "Bridgestone",
"model": "Potenza Sport",
"tireSpecs": "225 45 R17",
"mevsim": "Yaz",
"fuelNote": "D",
"performance": "Orta",
"price": 2568
},
{
"id": 3,
"model_id": 1,
"brand": "Michelin",
"model": "Pilot Alpin",
"tireSpecs": "225 45 R17",
Console
Postbot
Runner


C:\Users\06dz0\OneDrive\Ma
ngrok (Ctrl+C to quit)
Build better APIs with ngrok. Early access: ngrok.com/early-access
Session Status online
Account Solast (Plan: Free)
Version 3.5.0
Region Europe (eu)
Latency -
Web Interface http://127.0.0.1:4040
Forwarding https://3788-5-176-232-225.ngrok-free.app -> http://localhost:8000
Connections ttl opn rt1 rt5 p50 p90
0 0 0.00 0.00 0.00 0.00


https://3788-5-176-232-225.ngrok-free.app/api/tires/
Gov Study Software Stuff YouTube ChatGPT a cute baby dragon... Lucid Chart Boards | Trello UML Class Diyagra... audiotool BeepBox Django Tutorial (170) Django Tutori...
Django REST framework
Tire List
Tire List
OPTIONS
GET
GET /api/tires/
HTTP 200 OK
Allow: GET, OPTIONS, POST
Content-Type: application/json
Vary: Accept
[
{
"id": 1,
"model_id": 1,
"brand": "Bridgestone",
"model": "Potenza Sport",
"tireSpecs": "225 45 R17",
"mevsim": "Kış",
"fuelNote": "C",
"performance": "Orta",
"price": 2172
},
{
"id": 2,
"model_id": 1,
"brand": "Bridgestone",
"model": "Potenza Sport",
"tireSpecs": "225 45 R17",
"mevsim": "Yaz",
"fuelNote": "D",
"performance": "Orta",
"price": 2568
},
{
"id": 3,
"model_id": 1,
"brand": "Michelin",
"model": "Pilot Alpin",
"tireSpecs": "225 45 R17",
"mevsim": "Kış",


https://3788-5-176-232-225.ngrok-free.app/api/tires/
Gov Study Software Stuff YouTube ChatGPT a cute baby dragon... Lucid Chart Boards | Trello UML Class Diyagra... audiotool BeepBox Django Tutorial (170) Django Tutori...
Django REST framework
"mevsim": "Yaz",
"fuelNote": "B",
"performance": "Yüksek",
"price": 4683
},
{
"id": 307,
"model_id": 68,
"brand": "Michelin",
"model": "Pilot Alpin",
"tireSpecs": "225 65 R17",
"mevsim": "Kış",
"fuelNote": "C",
"performance": "Yüksek",
"price": 7506
}
]
Media type:
application/json
Content:
POST


https://3788-5-176-232-225.ngrok-free.app/api/tires/16/
Mail
Socials
Gov
Study
Software Stuff
YouTube
ChatGPT
a cute baby dragon...
Lucid Chart
Boards | Trello
UML Class Diyagra...
audiotool
BeepBox
Django Tutorial
(170) Django Tutori...
(2) WhatsApp
Django REST framework
Tire List / Tire Detail
Tire Detail
DELETE
OPTIONS
GET
GET /api/tires/16/
HTTP 200 OK
Allow: GET, OPTIONS, DELETE, PUT
Content-Type: application/json
Vary: Accept
{
"id": 16,
"model_id": 3,
"brand": "Michelin",
"model": "Pilot Sport 4",
"tireSpecs": "225 40 R18",
"mevsim": "Yaz",
"fuelNote": "B",
"performance": "Yüksek",
"price": 5999
}
Media type:
application/json
Content:




```
[
    {
        "id": 1,
        "model_id": 1,
        "brand": "Bridgestone",
        "model": "Potenza Sport",
        "tireSpecs": "225 45 R17",
        "mevsim": "Kış",
        "fuelNote": "C",
        "performance": "Orta",
        "price": 2172
    },
    {
        "id": 2,
        "model_id": 1,
        "brand": "Bridgestone",
        "model": "Potenza Sport",
        "tireSpecs": "225 45 R17",
        "mevsim": "Yaz",
        "fuelNote": "D",
        "performance": "Orta",
        "price": 2568
    },
    {
        "id": 3,
        "model_id": 1,
        "brand": "Michelin",
        "model": "Pilot Alpin",
        "tireSpecs": "225 45 R17",
        "mevsim": "Kış",
        "fuelNote": "C",
        "performance": "Orta",
        "price": 3474
    },
    {
        "id": 4,
        "model_id": 1,
        "brand": "Michelin",
        "model": "Pilot Alpin",
        "tireSpecs": "225 45 R17",
        "mevsim": "Yaz",
        "fuelNote": "C",
        "performance": "Orta",
        "price": 2914
    },
    {
        "id": 5,
        "model_id": 1,
        "brand": "Michelin",
        "model": "Pilot Alpin",
        "tireSpecs": "225 45 R17",
        "mevsim": "Dört Mevsim",
        "fuelNote": "C",
        "performance": "Orta",
        "price": 4696
    },
    {
        "id": 6,
        "model_id": 2,
        "brand": "Bridgestone",
        "model": "Blizzak",
        "tireSpecs": "205 55 R16",
        "mevsim": "Kış",
        "fuelNote": "C",
        "performance": "Orta",
        "price": 2561
    },
    {
        "id": 7,
        "model_id": 2,
```

https://3788-5-176-232-225.ngrok-free.app/admin/login/?next=/admin/

Mail | Socials | Gov | Study | Software Stuff | YouTube | ChatGPT | a cute baby dragon... | Lucid Chart | Boards | Trello | UML Class Diyagra... | audiotool | BeepB

Forbidden (403)

CSRF verification failed. Request aborted.

Help

Reason given for failure:

```
Origin checking failed - https://3788-5-176-232-225.ngrok-free.app does not match any trusted origins.
```

In general, this can occur when there is a genuine Cross Site Request Forgery, or when Django's CSRF mechanism has not been used correctly. For POST forms, you need to ensure:

- Your browser is accepting cookies.
- The view function passes a `request` to the template's `render` method.
- In the template, there is a `{% csrf_token %}` template tag inside each POST form that targets an internal URL.
- If you are not using `CsrfViewMiddleware`, then you must use `csrf_protect` on any views that use the `csrf_token` template tag, as well as those that accept the POST data.
- The form has a valid CSRF token. After logging in in another browser tab or hitting the back button after a login, you may need to reload the page with the form, because the token is rotated after a login.

You're seeing the help section of this page because you have `DEBUG = True` in your Django settings file. Change that to `False`, and only the initial error message will be displayed.

You can customize this page using the CSRF_FAILURE_VIEW setting.