

partAdvisor

Caner Ayrancı



part Advisor

Araç sahiplerine araçları ve kullanım alışkanlıklarına uygun şekilde lastik ve ilerde 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 yapabilmektedir.
-



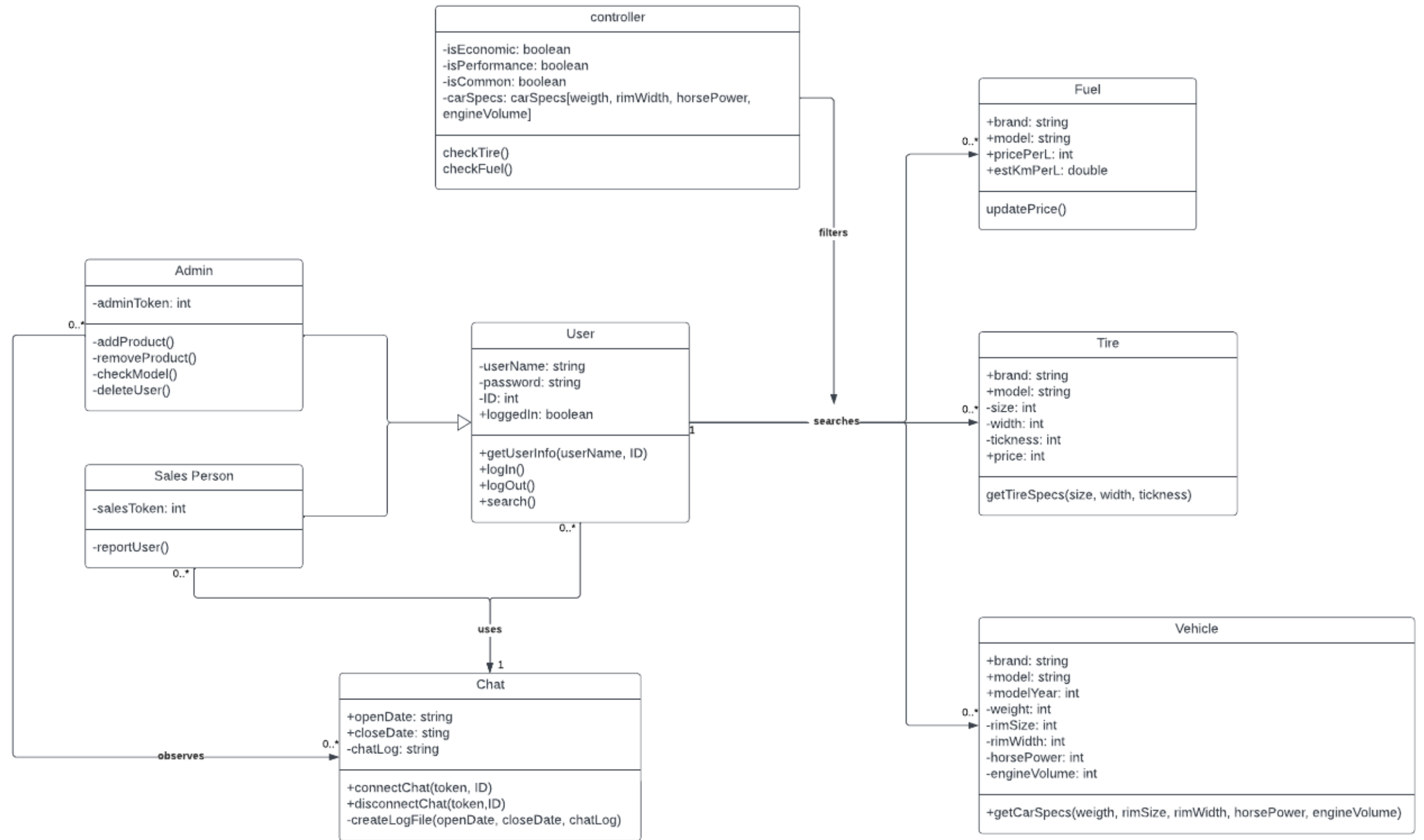
partAdvisor

Diyagramlar

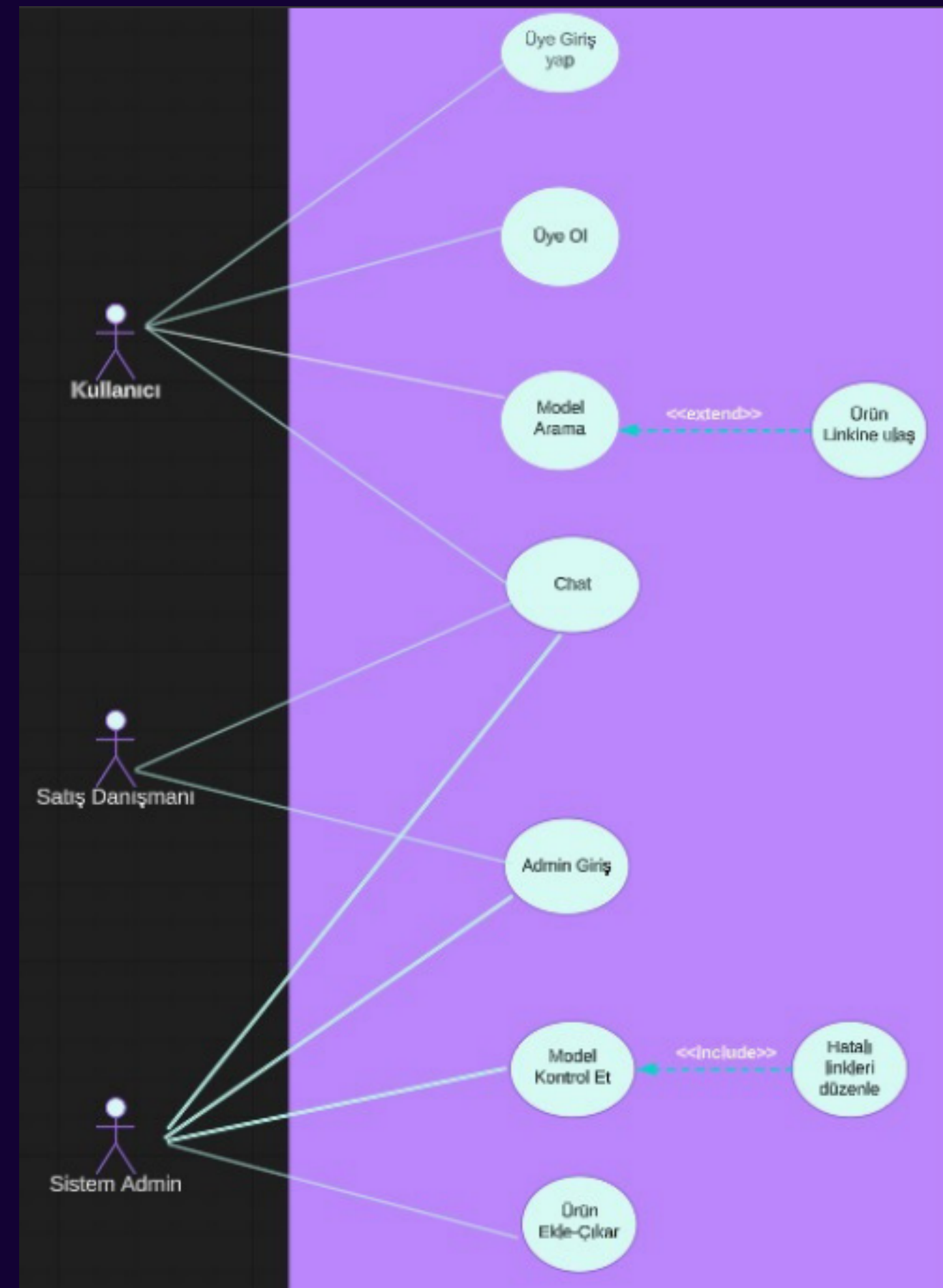
- Class Diyagramı
- Use - Case Diyagramı
- Sequence Diyagramı
- ER Diyagramı



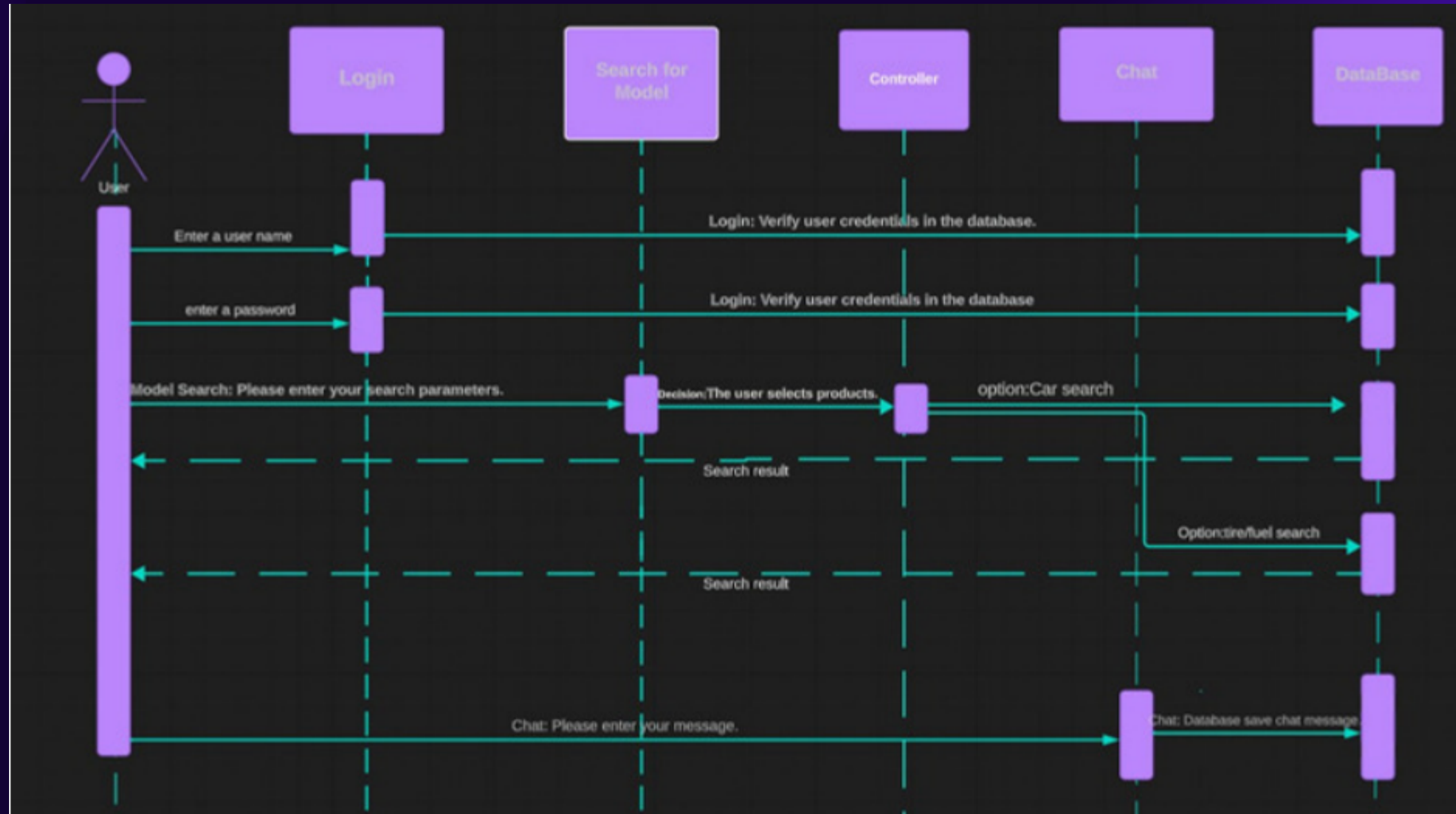
Class Diyagramı



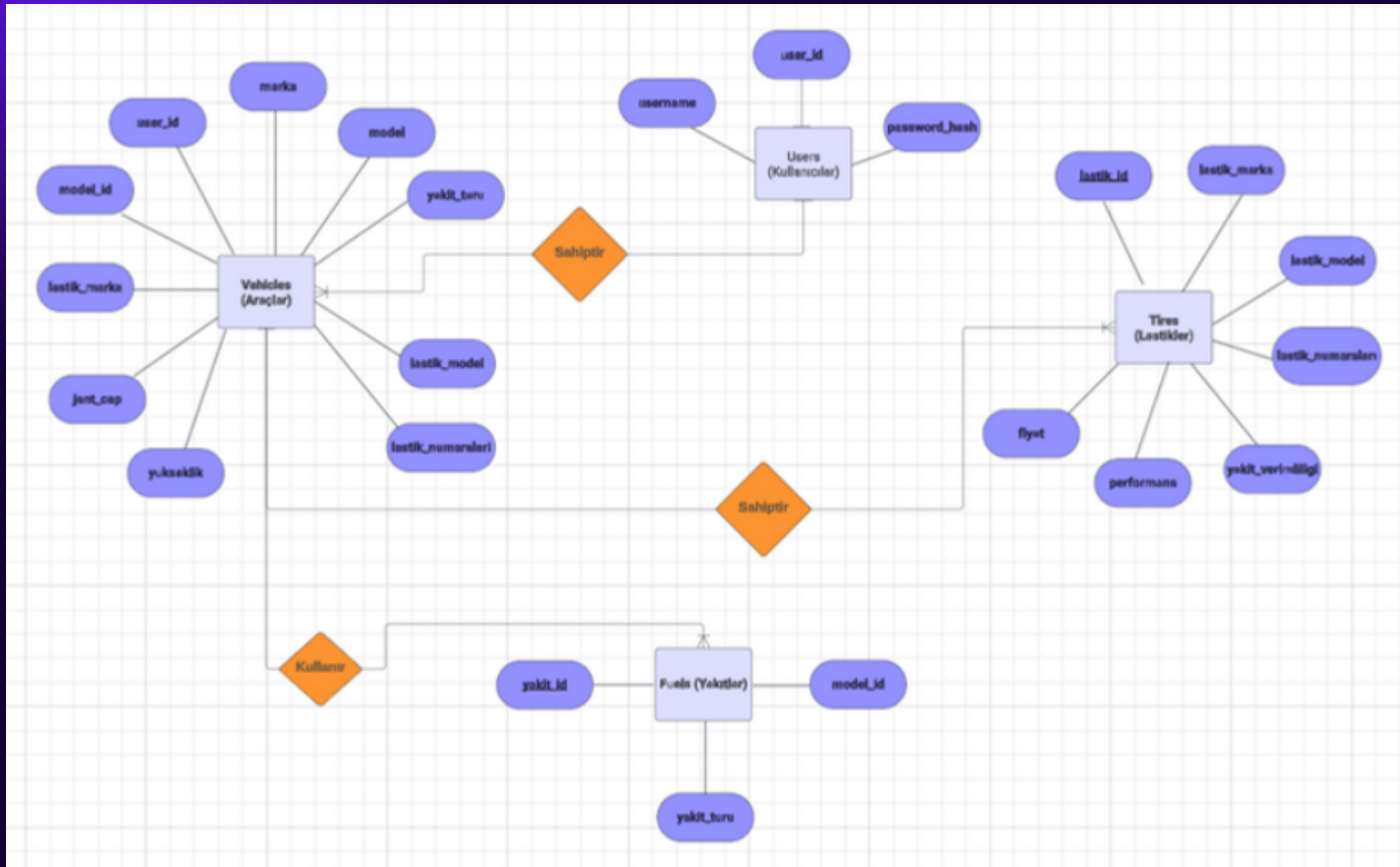
Use - Case Diyagramı



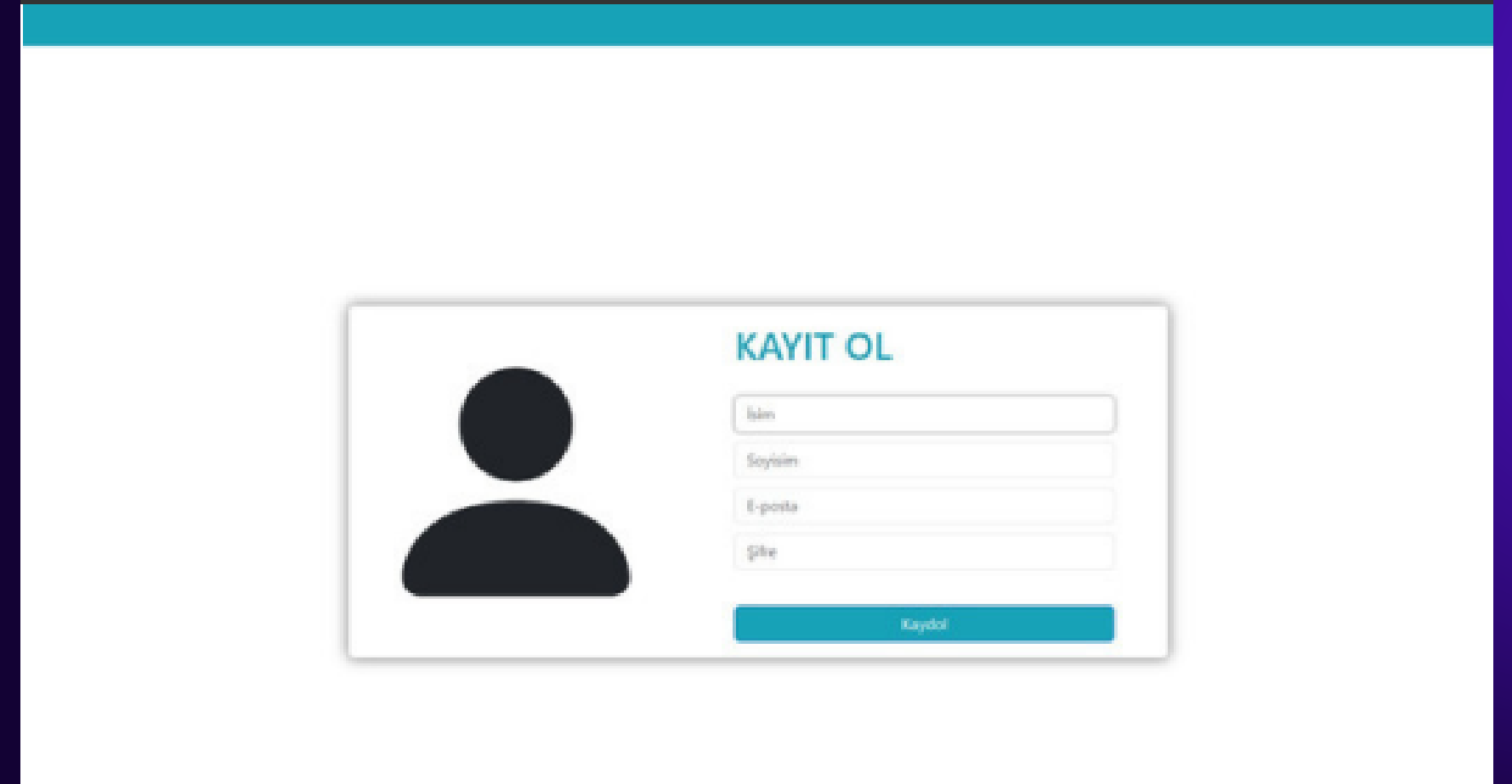
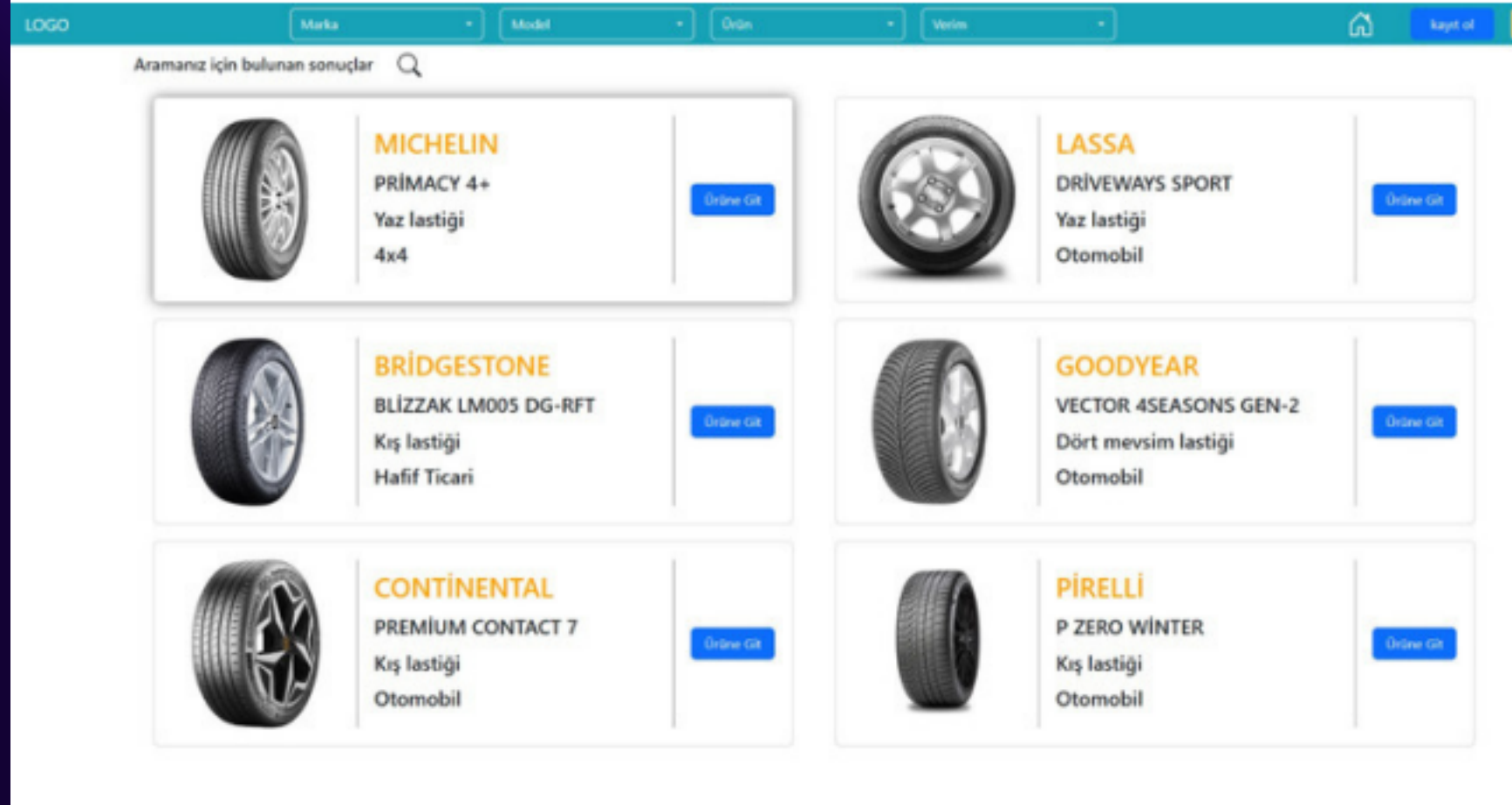
Sequence Diyagramı



ER Diyagramı



Demo UI



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
framework'ün oluşturacağı sqlite3 veritabanında tutuluyor

```
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
```

```
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
```

```
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

```
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

```
@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)
```

```
@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

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

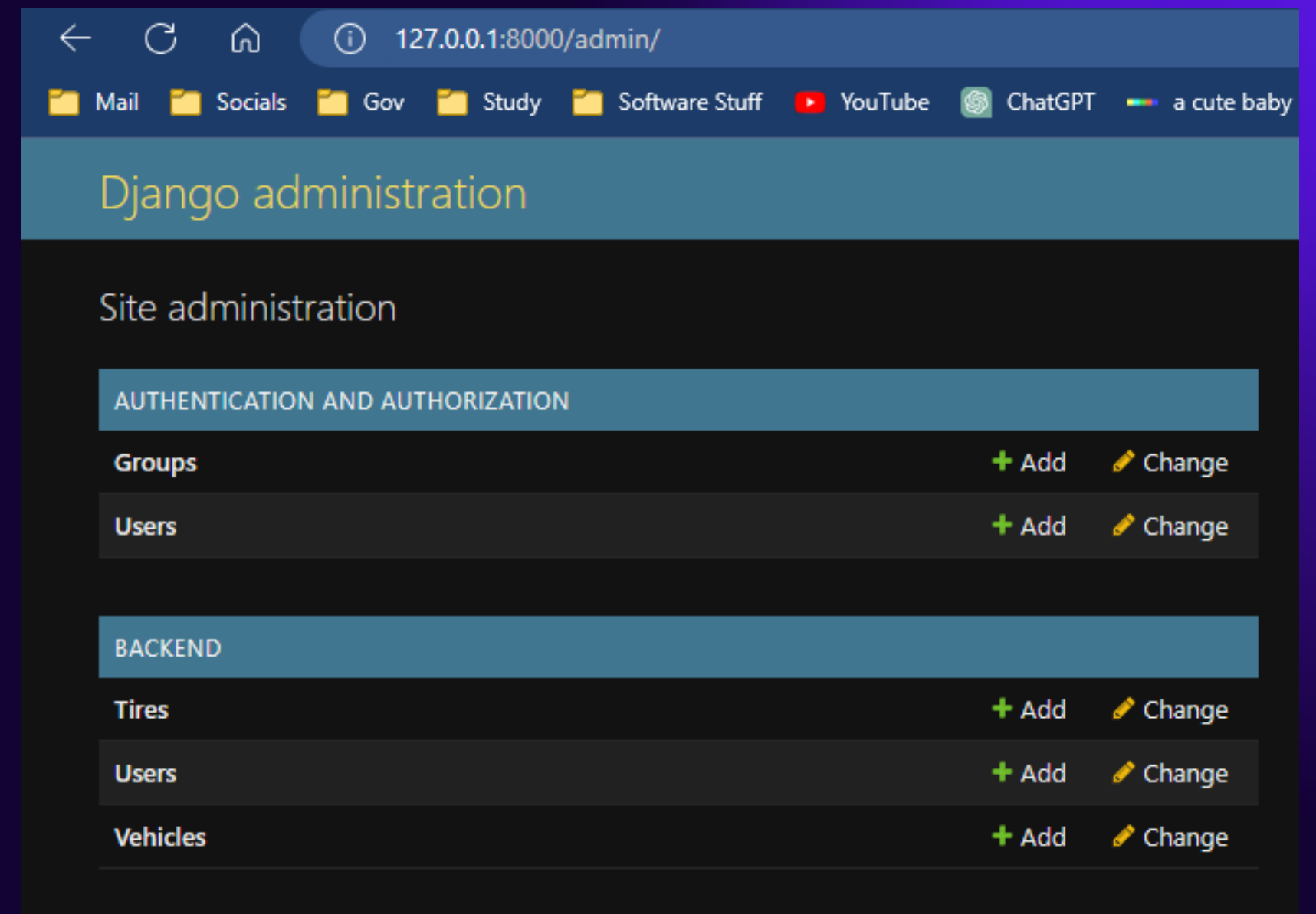
class TireSerializer(serializers.ModelSerializer):
    class Meta:
        model = Tire
        fields = ['id', 'model_id', 'brand', 'model', 'tireSpecs', 'mevsim', 'fuelNote', 'performance', 'price']

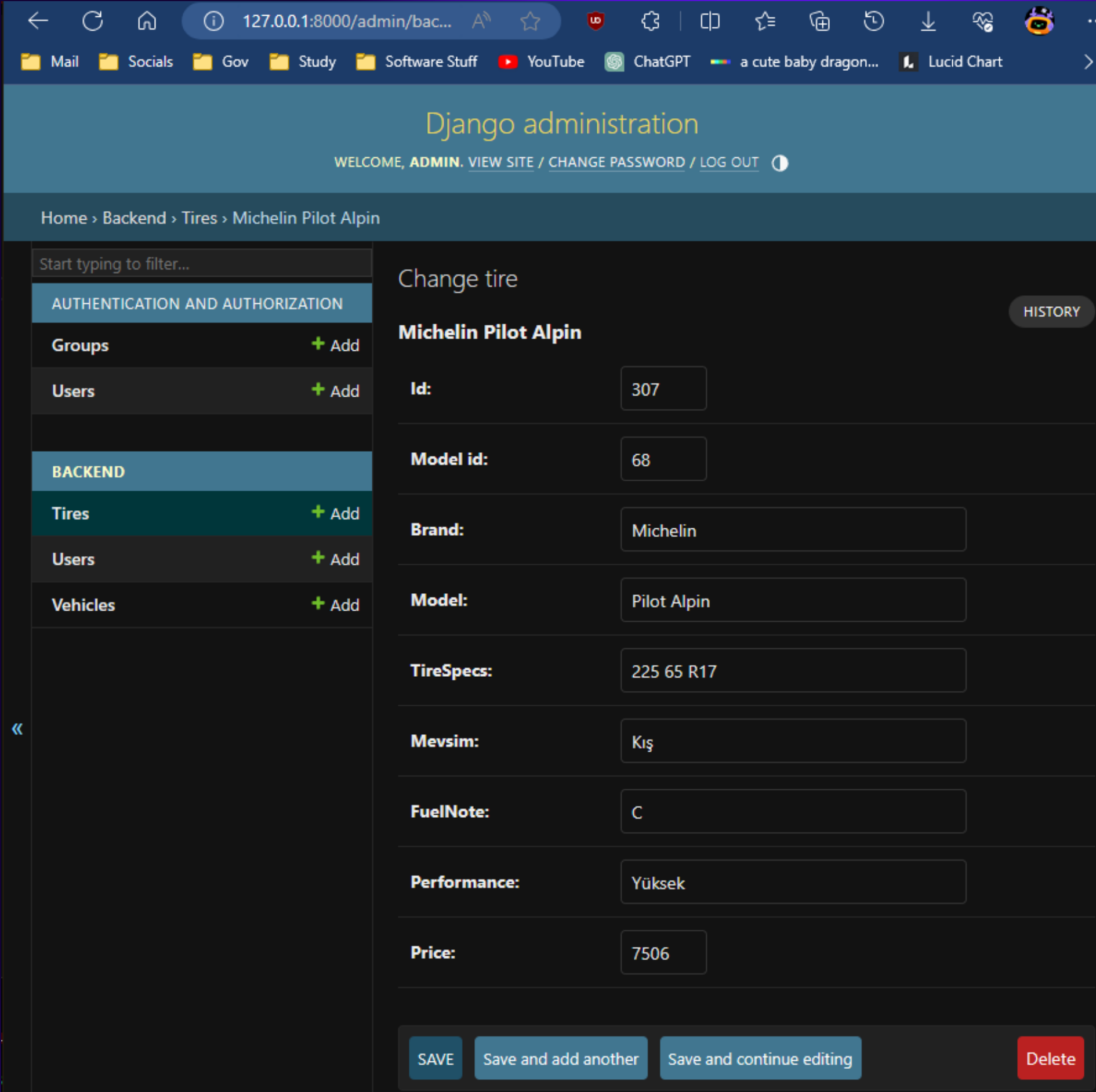
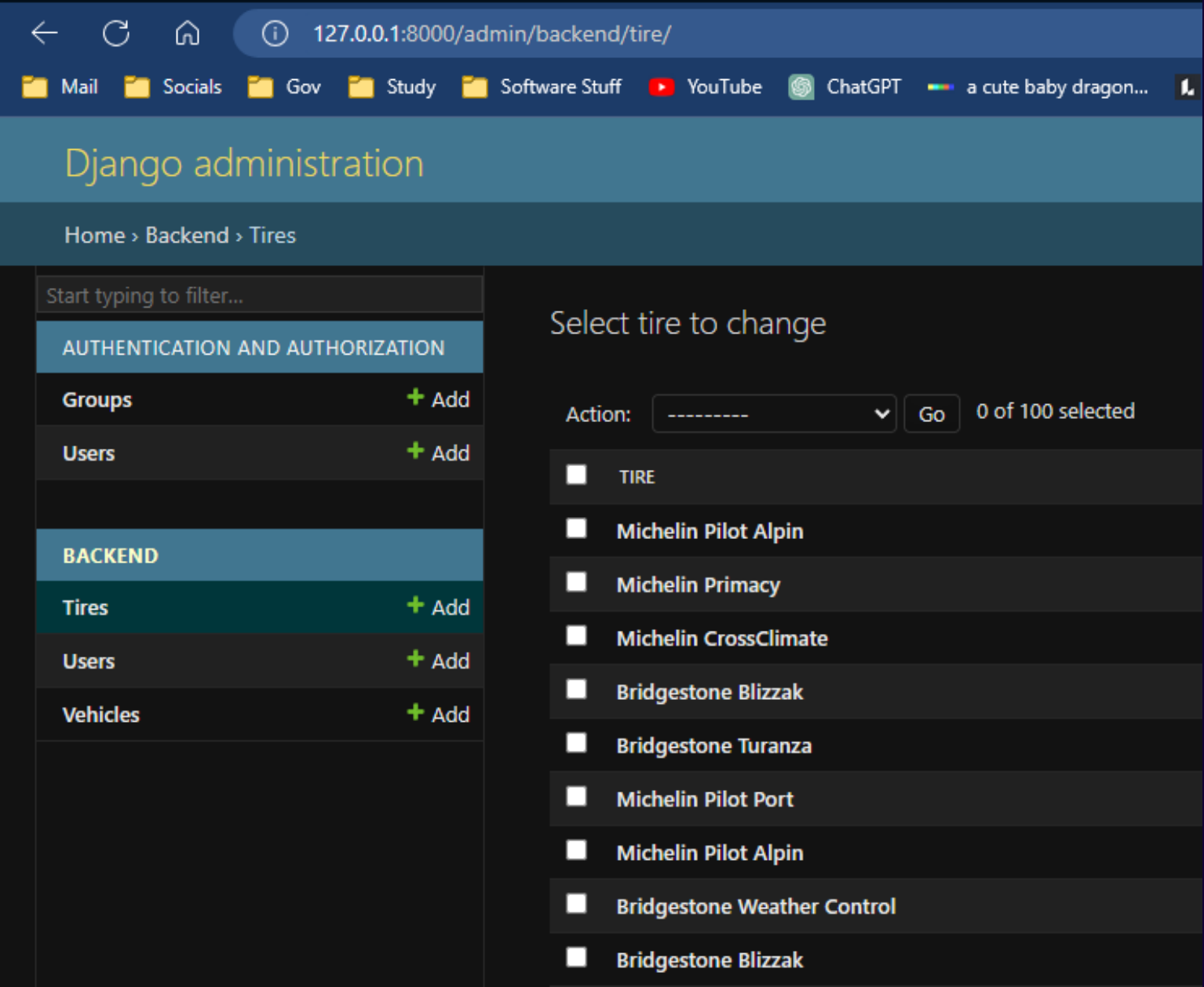
class VehicleSerializer(serializers.ModelSerializer):
    class Meta:
        model = Vehicle
        fields = ['model_id', 'user_id', 'brand', 'model', 'fuelType', 'tireSpecs', 'tireWidth', 'tireHeight', 'rimSize']
```

/admin

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

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






```
[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://127.0.0.1:8000/api/tires/ 🔴

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

1

Body

Cookies

Headers (10)

Test Results

🌐 200 OK 15 ms 47.94 KB

📄 Save as example

⋮

Pretty

Raw

Preview

Visualize

JSON ⌵

↺

📄

🔍

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

[

{

"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"

📄

✓

🔍

📄 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/

GovStudySoftware StuffYouTubeChatGPTa cute baby dragon...Lucid ChartBoards | TrelloUML Class Diyagra...audiotoolBeepBoxDjango Tutorial(170) Django Tutori...WhatsApp

Django REST framework

Tire List

Tire List

OPTIONSGET

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",
    "fuelNote": "C",
    "performance": "Yüksek",
    "price": 4683
  }
]
```

https://3788-5-176-232-225.ngrok-free.app/api/tires/

GovStudySoftware StuffYouTubeChatGPTa cute baby dragon...Lucid ChartBoards | TrelloUML Class Diyagra...audiotoolBeepBoxDjango Tutorial(170) Django Tutori...WhatsApp

Django REST framework

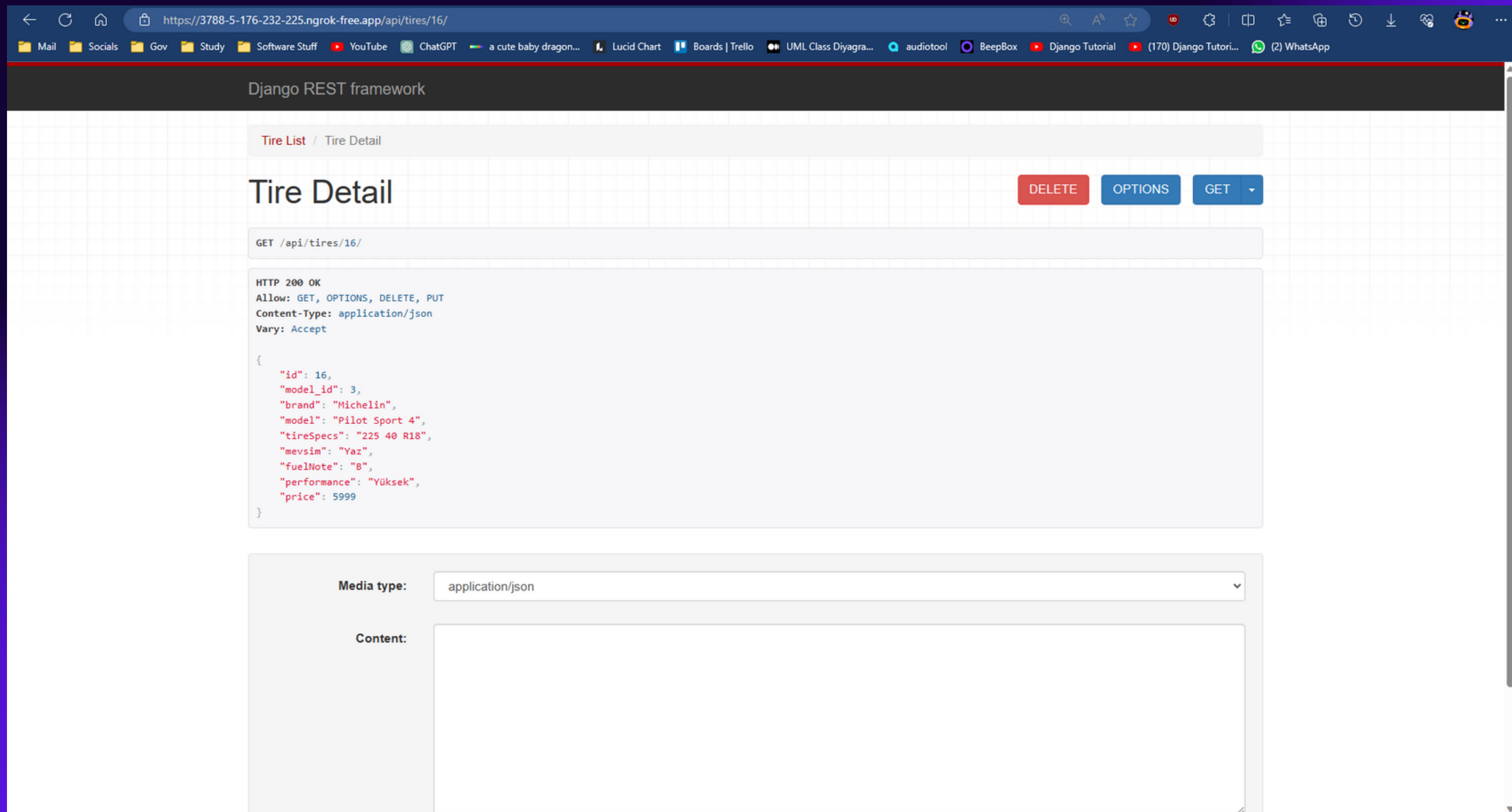
```
tireSpecs": "225 45 R17",
"mevsim": "Yaz",
"fuelNote": "B",
"performance": "Yüksek",
"price": 4683
},
{
  "id": 307,
  "model_id": 68,
  "brand": "Michelin",
  "model": "Pilot Alpin",
  "tireSpecs": "225 65 R17",
  "mevsim": "Kış",
  "fuelNote": "C",
  "performance": "Yüksek",
  "price": 7506
}
]
```

Media type:

application/json

Content:

POST



← ↻ 🏠 🔒 https://3788-5-176-232-225.ngrok-free.app/api/tires.json

📁 Mail 📁 Socials 📁 Gov 📁 Study 📁 Software Stuff 📺 YouTube 🗯 ChatGPT

```
1  [
2    {
3      "id": 1,
4      "model_id": 1,
5      "brand": "Bridgestone",
6      "model": "Potenza Sport",
7      "tireSpecs": "225 45 R17",
8      "mevsim": "Kış",
9      "fuelNote": "C",
10     "performance": "Orta",
11     "price": 2172
12   },
13   {
14     "id": 2,
15     "model_id": 1,
16     "brand": "Bridgestone",
17     "model": "Potenza Sport",
18     "tireSpecs": "225 45 R17",
19     "mevsim": "Yaz",
20     "fuelNote": "D",
21     "performance": "Orta",
22     "price": 2568
23   },
24   {
25     "id": 3,
26     "model_id": 1,
27     "brand": "Michelin",
28     "model": "Pilot Alpin",
29     "tireSpecs": "225 45 R17",
30     "mevsim": "Kış",
31     "fuelNote": "C",
32     "performance": "Orta",
33     "price": 3474
34   },
35   {
36     "id": 4,
37     "model_id": 1,
38     "brand": "Michelin",
39     "model": "Pilot Alpin",
40     "tireSpecs": "225 45 R17",
41     "mevsim": "Yaz",
42     "fuelNote": "C",
43     "performance": "Orta",
44     "price": 2914
45   },
46   {
47     "id": 5,
48     "model_id": 1,
49     "brand": "Michelin",
50     "model": "Pilot Alpin",
51     "tireSpecs": "225 45 R17",
52     "mevsim": "Dört Mevsim",
53     "fuelNote": "C",
54     "performance": "Orta",
55     "price": 4696
56   },
57   {
58     "id": 6,
59     "model_id": 2,
60     "brand": "Bridgestone",
61     "model": "Blizzak",
62     "tireSpecs": "205 55 R16",
63     "mevsim": "Kış",
64     "fuelNote": "C",
65     "performance": "Orta",
66     "price": 2561
67   },
68   {
69     "id": 7,
70     "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 🎧 Beepf

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.

Tasarım Desenleri

```
1 import VehilePartData from "./VehiclePartData";
2 class PartFlyweightFactory {
3   constructor() {
4     this.flyweights = {};
5   }
6
7   getFlyweight(intrinsicData) {
8     const key = JSON.stringify(intrinsicData);
9     if (!this.flyweights[key]) {
10       this.flyweights[key] = intrinsicData;
11     }
12     return this.flyweights[key];
13   }
14
15   getFlyweightsCount() {
16     return Object.keys(this.flyweights).length;
17   }
18 }
19
20 const VehiclePartItem = ({ partData }) => {
21   return (
22     <div className="item">
23       <div className="image">
24         <img width={200} height={200} src={partData.resim} alt="ürün resmi" />
25       </div>
26
27       <div className="information">
28         <h2>{partData.marka}</h2>
29         <h3>{partData.model}</h3>
30         <h4>{partData.aractipi}</h4>
31         <h5>{partData.ozelligi}</h5>
32       </div>
33       <div className="bbut">
34         <a href={partData.link} target="_blank" rel="noopener noreferrer">
35           <button className="btn btn-primary but">ürüne git</button>
36         </a>
37       </div>
38     </div>
39   );
40 };
41
42 const FlyweightVehicleParts = (props) => {
43   const deger = props.dropdown;
44   const flyweightFactory = new PartFlyweightFactory();
45
46   return (
47     <div>
48       <div className="cards">
49         {VehilePartData.filter((part) => {
50           return deger.toLowerCase() === ""
51             ? part
52             : part.aractipi.toLowerCase().includes(deger);
53         }).map((part) => (
54           <VehiclePartItem
55             key={part.id}
56             partData={flyweightFactory.getFlyweight({
57               marka: part.marka,
58               model: part.model,
59               aractipi: part.aractipi,
60               ozelligi: part.ozelligi,
61               link: part.link,
62               resim: part.resim,
63               type: part.aractipi,
64             })}
65           </>
66         ))}
67       </div>
68     </div>
69   );
70 };
71
```

Flyweight Desing Pattern

```

1 import VehilePartData from "./VehiclePartData";
2 class PartFlyweightFactory {
3   constructor() {
4     this.flyweights = {};
5   }
6
7   getFlyweight(intrinsicData) {
8     const key = JSON.stringify(intrinsicData);
9     if (!this.flyweights[key]) {
10       this.flyweights[key] = intrinsicData;
11     }
12     return this.flyweights[key];
13   }
14
15   getFlyweightsCount() {
16     return Object.keys(this.flyweights).length;
17   }
18 }
19
20 const VehiclePartItem = ({ partData }) => {
21   return (
22     <div className="item">
23       <div className="image">
24         <img width={200} height={200} src={partData.resim} alt="ürün resmi" />
25       </div>
26
27       <div className="information">
28         <h2>{partData.marka}</h2>
29         <h3>{partData.model}</h3>
30         <h4>{partData.aractipi}</h4>
31         <h5>{partData.ozelligi}</h5>
32       </div>
33       <div className="bbut">
34         <a href={partData.link} target="_blank" rel="noopener noreferrer">
35           <button className="btn btn-primary but">ürüne git</button>
36         </a>
37       </div>
38     </div>
39   );
40 };
41
42 const FlyweightVehicleParts = (props) => {
43   const deger = props.dropdown;
44   const flyweightFactory = new PartFlyweightFactory();
45
46   return (
47     <div>
48       <div className="cards">
49         {VehilePartData.filter((part) => {
50           return deger.toLowerCase() === ""
51             ? part
52             : part.aractipi.toLowerCase().includes(deger);
53         }).map((part) => (
54           <VehiclePartItem
55             key={part.id}
56             partData={flyweightFactory.getFlyweight({
57               marka: part.marka,
58               model: part.model,
59               aractipi: part.aractipi,
60               ozelligi: part.ozelligi,
61               link: part.link,
62               resim: part.resim,
63               type: part.aractipi,
64             })}
59         </div>
60       </div>
61     </div>
62   );
63 };
64
65

```

Strategy Desing Pattern

Proxy Desing Pattern

Login ve Register işlemleri için
firebase yapısı proxy olarak
kullanılıp, işlemler back-end ve
front-end dışında handle
edilmektedir.

```
// Import the functions you need from the SDKs you need
import { getAuth } from "firebase/auth";
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyDRutDFSGaq_1jsBN0ca0053ak6CJjuHwI",
  authDomain: "fir-bb645.firebaseio.com",
  projectId: "fir-bb645",
  storageBucket: "fir-bb645.appspot.com",
  messagingSenderId: "330369797065",
  appId: "1:330369797065:web:faad64795503068491b618",
  measurementId: "G-04E6D7DRMV",
};

// Initialize Firebase
export const app = initializeApp(firebaseConfig);
export const analytics = getAnalytics(app);
export const auth = getAuth(app);
```