

# Stajyer Elemesi İçin Backend Uygulaması

## Görevi

### Görev Hedefleri:

- Go (veya Python) ile temel API servisi yazma becerilerinin test edilmesi.
  - PostgreSQL veritabanını SQL seviyesinde programlama becerilerinin test edilmesi.
  - Veri filtreleme, sıralama ve sayfalama işlevselliğinin oluşturulması.
  - Temiz ve okunaklı kod yazma yeteneğinin değerlendirilmesi.
  - Büyük bir veri tabanını sağlıklı ve hızlı bir şekilde işleyen kod yazım yeteneğinin değerlendirilmesi.
- Python kullanarak API oluşturulmuştur.
  - İstek gövdeleri aracılığıyla sıralama ve URL aracılığıyla sayfalama işlevselliğinin oluşturulması.

### Görev gereksinimleri:

- Go veya Python web kütüphanesi/sistemi kullanarak bir API servisi oluşturun. ✓
- Oluşturduğunuz servis aşağıdaki temel işlevselliği sunmalıdır:
- İstekleri almak için bir uç nokta (endpoint) ( /assignment/query )
- oluşturun. ✓

`@app.route('/assignment/query', methods=['POST'])`

- Gelen isteği doğru bir şekilde işleyin, verileri filtreleyin, sıralayın
- ve sayfalayın. ✓

#### İstek:

- URL: [http://127.0.0.1:5000/assignment/query?page=1&per\\_page=20](http://127.0.0.1:5000/assignment/query?page=1&per_page=20)
- HTTP Metodu: POST
- İstek Başlığı: Content-Type: application/json

- Servis kullanıcıların gönderdiği filtreler ve sıralama isteklerini kabul etmeli
- ve sonuçları bu kriterlere göre döndürmelidir. ✓

```
{
  "filters": {"main_af_vcf": 1},
  "ordering": [{"main_af_vcf": "ASC"}]
}
```

- Verileri döndürmek için size bu test ile iletilmiş olan SQL dosyasıyla oluşturabilirsiniz. ✓
- İsteklerin doğru bir şekilde işlenmesi ve hataların uygun şekilde yönetilmesi gereklidir. Eğer istemci tarafından gönderilmiş istek aşağıda belirtilen biçime uygun değilse "400 - Bad Request" cevabı döndürülmelidir. ✓

```
{
  "error": "Internal Server Error: 400 Bad Request: The browser (or proxy) sent a request that this server could not understand."
}
```