

APILER VE ÇIKTILARI

Advisors

- > **GET** api/Advisors
- > **POST** api/Advisors
- > **GET** api/Advisors/{id}
- > **PUT** api/Advisors/{id}
- > **DEL** api/Advisors/{id}

```
[
  {
    "advisorID": 1,
    "fullname": "Prof. Dr. Hamdi Tolga Kahraman",
    "title": "Bölüm Başkanı",
    "department": "Yazılım Mühendisliği Bölümü",
    "email": "tolgakahraman@ktu.edu.tr",
    "students": [
      {
        "studentID": 10001,
        "firstname": "Ahmet",
        "lastname": "Yılmaz"
      },
      {
        "studentID": 10002,
        "firstname": "Mehmet",
        "lastname": "Kaya"
      },
      {
        "studentID": 10003,
        "firstname": "Ayşe",
        "lastname": "Demir"
      }
    ]
  }
]
```

Students

- > **GET** api/Students
- > **POST** api/Students
- > **GET** api/Students/{id}
- > **PUT** api/Students/{id}
- > **DEL** api/Students/{id}

```
[
  {
    "studentID": 10001,
    "firstname": "Ahmet",
    "lastname": "Yılmaz",
    "email": "ahmet.yilmaz@ogr.ktu.edu.tr",
    "advisor": {
      "fullname": "Prof. Dr. Hamdi Tolga Kahraman",
      "title": "Bölüm Başkanı",
      "department": "Yazılım Mühendisliği Bölümü"
    },
    "courses": [
      {
        "courseID": 1,
        "courseName": "Algoritma ve Programlama"
      },
      {
        "courseID": 2,
        "courseName": "Veritabanı Yönetim Sistemleri"
      },
      {
        "courseID": 8,
        "courseName": "İşletim Sistemleri"
      }
    ]
  }
]
```

Users

- > **GET** api/Users
- > **GET** api/Users{id}
- > **PUT** api/Users{id}
- > **DEL** api/Users{id}
- > **POST** api/Users

```
[
  {
    "userID": 11,
    "username": "tolgakahraman",
    "passwordHash": "hashedpassword1",
    "role": "Advisor",
    "relatedID": 1,
    "email": "tolgakahraman@ktu.edu.tr"
  },
  {
    "userID": 12,
    "username": "hozyurt",
    "passwordHash": "hashedpassword2",
    "role": "Advisor",
    "relatedID": 2,
    "email": "hozyurt@ktu.edu.tr"
  }
],
```

Courses

- > **GET** api/Courses
- > **POST** api/Courses
- > **GET** api/Courses/{id}
- > **PUT** api/Courses/{id}
- > **DEL** api/Courses/{id}

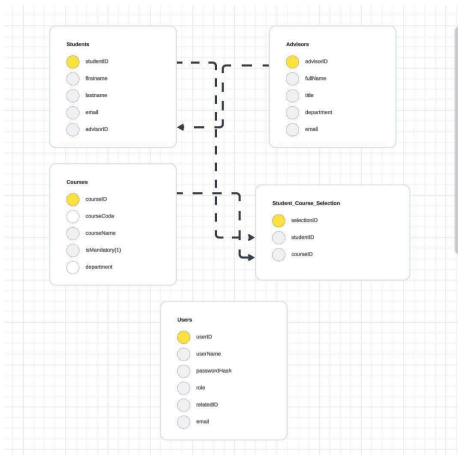
```
[
  {
    "courseID": 1,
    "courseCode": "YAZ101",
    "courseName": "Algoritma ve Programlama",
    "isMandatory": true,
    "credit": 4,
    "department": "Yazılım Mühendisliği",
    "studentCourseSelections": [
      {
        "studentID": 10001
      },
      {
        "studentID": 10002
      },
      {
        "studentID": 10003
      },
      {
        "studentID": 10004
      }
    ]
  }
],
```

StudentCourseSelection

```
> GET api/StudentCourseSelection
> POST api/StudentCourseSelection
> GET api/StudentCourseSelection{id}
> PUT api/StudentCourseSelection{id}
> DEL api/StudentCourseSelection{id}
```

```
{
  {
    "studentID":10001,
    "firstname":"Ahmet",
    "lastname":"Yılmaz",
    "courses":[
      {
        "courseID":1,
        "courseName":"Algoritma ve Programlama"
      },
      {
        "courseID":2,
        "courseName":"Veritabanı Yönetim Sistemleri"
      },
      {
        "courseID":8,
        "courseName":"İşletim Sistemleri"
      },
      {
        "courseID":14,
        "courseName":"Bitirme Projesi 1"
      }
    ]
  }
}
```

E-R DİYAGRAMI



E-R Diyagramı Açıklamalar:

Student - Advsors İlişkisi

- Açıklama: Her öğrenciye(students) bir danışman (advisors) atanmıştır. Bu da *Students* tablosundaki advisorID ile *Advisors* tablosundaki advisorID arasında bir bağlantı oluşturur.
- Tür:Bire çok (One-to-Many):Bir danışman birden çok öğrenciye danışmanlık yapabilir ancak bir öğrenciye yalnızca bir danışmana bağlıdır.

Student - Courses İlişkisi

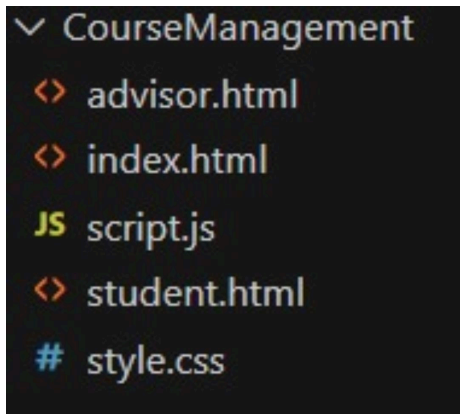
- Açıklama: Öğrenciler (students) ders (courses) seçimi yapabilirler. Bu ilişki StudentCourseSelection ile modellenmiştir.

- Tür: Çoktan çoğa (Many-to-Many): Bir öğrenci(students) birden fazla ders (courses) seçebilir. Ayrıca bir derse de (courses) birden fazla öğrenci (students) kayıt olabilir.

Users

- Açıklama: Users giriş yapan kullanıcıyı belirlemek amacıyla oluşturulmuştur. İlişkisi yoktur. Kullanıcı giriş esnasında danışman(advisors) veya öğrenci(students) olarak giriş yapabilir.

FRONTEND BÖLÜMÜ



Açıklamalar:

index.html: Giriş sayfasıdır. Kullanıcı giriş rolünü (danışman veya öğrenci) seçip gerekli bilgileri girdikten sonra kullanıcıyı danışman(*advisor.html*) veya öğrenci(*student.html*) sayfasına yönlendirir.

advisor.html: Danışman(advisor) sayfasını oluşturur. Danışman bu sayfada öğrencinin(student) ders seçimini onaylayabilir. Ayrıca öğrenci bilgilerini de görebilir.

student.html: Öğrenciye atanan dersleri ve yeni dersleri listeler. Daha sonra dersleri gönder butonuyla dersleri danışman onayına gönderir.

script.js: Öğrenci bilgileri, danışman bilgileri ve ders bilgileri gibi backend'den alınan verilerin frontend'e aktarılmasını sağlar.

style.css: Web sayfasının görsel tasarımını kontrol etmek için kullanılır.