

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİL 372 - Veritabanı Sistemleri**

**Proje Final Raporu**

**PROJE İSMİ**



**GRUP ÜYELERİ**

| **AD - SOYAD** | **NUMARA** | **OKUL MAİLİ** |
| --- | --- | --- |
| **Ahmet Karamercan** | **191101017** | **akaramercan@etu.edu.tr** |
| **Berkay Sarı** | **191101035** | **berkaysari@etu.edu.tr** |

**PROJE TANIM VE GEREKSİNİMLERİ**

Projemizde yerli şirketlerimizden olan “Udemy” tarzında bir web sitesi tasarladık. Bu web sitesi, kullanıcıların ilgili olduğu alanlara göre çeşitli kurslardan faydalanmasını sağlayan online bir video eğitim platformu görevi görmektedir. Sitemizde sisteme kurs yükleyebilen, kurstaki her bir farklı derse ait sorulan sorulara, o soru başlığı altında cevap verebilen eğitmenlerin yanı sıra bu dersleri izleyebilen, sorular sorabilen, kurslara yorum ve puan bırakabilen öğrenciler olmak üzere iki farklı kullanıcı bulunmaktadır. Her öğrenci kendisine ait aldığı dersler, profil fotoğrafı ve kişisel veriler gibi birçok bilgi içeren profil sayfasına sahiptir. Aynı şekilde eğitmenler de bilgi sahibi olmadığı farklı kategorilerdeki kurslara kayıt olmakta özgürlerdir, onların da profil ve kurslarım sayfası mevcuttur.

Veri tabanında bulunan ‘date’ gibi değiştirilmesi mantıksız nitelikler dışındaki her attribute update edilebilmektedir (email ve şifre dahil). Update ve yeni tuple işlemlerindeki kısıtların bir kısmı kodlarda bulunan conditional statment blockları ile kontrol edilirken bir kısmı ise veri tabanında bulunan constraint’ler ile kontrol edilmektedir.

Sitemizde kategori ve isim üzerinden kursları arayan bir arama alanı bulunmaktadır. Aranan kelime ile eşleşen veya o kelimeyi içeren kategori veya isme sahip kursları sonuc olarak göstermektedir.

Her kursun kendine ait dersleri düzenleyebileceği sayfada derslerin başlığı, açıklaması, video url’si update edilebilmektedir. Kursa ait diğer bilgiler ise başka bir sayfada update edilmektedir.

Kullanıcı şifreleri güvenlik ve gizlilik açısından veritabanında hash’lenmiş şekilde tutulmaktadır. Hash fonksiyonu olarak “SHA-256” metodu kullanılmıştır.

**KAVRAMSAL TASARIM**

# User

Relation Schema: User(id, email, password, first\_name, last\_name, date, role, unvan, adres, telefon, github, twitter, instagram, linkedin, image\_file)

Candidate Keys: {id}, {email}

Primary Key: {id}

Foreign Key: None

Attribute Explanations and Domains:

*"id" INTEGER NOT NULL,*

*"email" VARCHAR(150) NOT NULL,*

*"password" VARCHAR(150) NOT NULL,*

*"first\_name" VARCHAR(50) NOT NULL,*

*"last\_name" VARCHAR(50) NOT NULL,*

*"date" DATETIME,*

*"role" INTEGER, ----> attribute-defined specialization için*

*"unvan" VARCHAR(40),*

*"adres" VARCHAR(50),*

*"telefon" VARCHAR(12),*

*"github" VARCHAR(40),*

*"twitter" VARCHAR(40),*

*"instagram" VARCHAR(40),*

*"linkedin" VARCHAR(40),*

*"image\_file" VARCHAR(20) NOT NULL*

# Egitmen

Relation Schema: Egitmen(id, ozgecmis, age)

Candidate Keys: {id}, {ozgecmis}

Primary Key: {id}

Foreign Key: {id} references user.id

Attribute Explanations and Domains:

*""id" INTEGER NOT NULL,*

*"ozgecmis" VARCHAR(1000),*

*"age" INTEGER*

# Kurs

Relation Schema: Kurs(id, isim, kategori, date, kisa\_aciklama, uzun\_aciklama, egitmen\_id, image\_file)

Candidate Keys: {id}, {isim}, {date}

Primary Key: {id}

Foreign Key: {egitmen\_id} references egitmen.id

Attribute Explanations and Domains:

*"id" INTEGER NOT NULL,*

*"isim" VARCHAR(200),*

*"kategori" VARCHAR(200),*

*"date" DATETIME,*

*"kisa\_aciklama" VARCHAR(100),*

*"uzun\_aciklama" VARCHAR(2000),*

*"egitmen\_id" INTEGER,*

*"image\_file" VARCHAR(20) NOT NULL*

# Subs

Relation Schema: Subs(user\_id, kurs\_id)

Candidate Keys: {user\_id, kurs\_id}

Primary Key: {user\_id, kurs\_id}

Foreign Key: {user\_id} references user.id, {kurs\_id} references kurs.id

Attribute Explanations and Domains:

*"user\_id" INTEGER,*

*"kurs\_id" INTEGER*

# Kurs\_Yorum

Relation Schema: Kurs\_Yorum(id, icerik, date, puan, user\_id, kurs\_id)

Candidate Keys: {user\_id, kurs\_id}

Primary Key: {user\_id, kurs\_id}

Foreign Key: {user\_id} references user.id, {kurs\_id} references kurs.id

Attribute Explanations and Domains:

*"id" INTEGER NOT NULL,*

*"icerik" VARCHAR(100),*

*"date" DATETIME,*

*"puan" INTEGER CHECK("puan" > 0 AND "puan" <= 5),*

*"user\_id" INTEGER NOT NULL,*

*"kurs\_id" INTEGER NOT NULL*

# Ders

Relation Schema: Ders(id, baslik, aciklama, video\_url, date, kurs\_id)

Candidate Keys: {id}, {date}

Primary Key: {id}

Foreign Key: {kurs\_id} references kurs.id

Attribute Explanations and Domains:

*"id" INTEGER NOT NULL,*

*"video\_url" VARCHAR(1000),*

*"baslik" VARCHAR(100) NOT NULL,*

*"aciklama" VARCHAR(100),*

*"date" DATETIME,*

*"kurs\_id" INTEGER*

# Ders\_Yorum

Relation Schema: Ders\_Yorum(id, baslik, icerik, date, user\_id, ders\_id, parent\_yorum)

Candidate Keys: {id}, {date}

Primary Key: {id}

Foreign Key: {user\_id} references user.id {ders\_id} references ders.id

{parent\_yorum} references ders\_yorum.id

Attribute Explanations and Domains:

*"id" INTEGER NOT NULL,*

*"baslik" VARCHAR(25),*

*"icerik" VARCHAR(100),*

*"date" DATETIME,*

*"user\_id" INTEGER NOT NULL,*

*"ders\_id" INTEGER NOT NULL,*

*"parent\_yorum" INTEGER*

**UYGULAMA AYRINTILARI**

Web uygulamasının back-end’i Flask çerçevesi altında Python 3'te kodlanmıştır. Front-end’te ise HTML, CSS, JavaScript ve Bootstrap kullanıldı. Ayrıca Jinja2 şablonlama dili front-end ve back-end arasındaki boşluğu doldurmak için kullanılır. Veritabanı Yönetim Sistemi olarak yerelde SQLite kullanılmıştır.

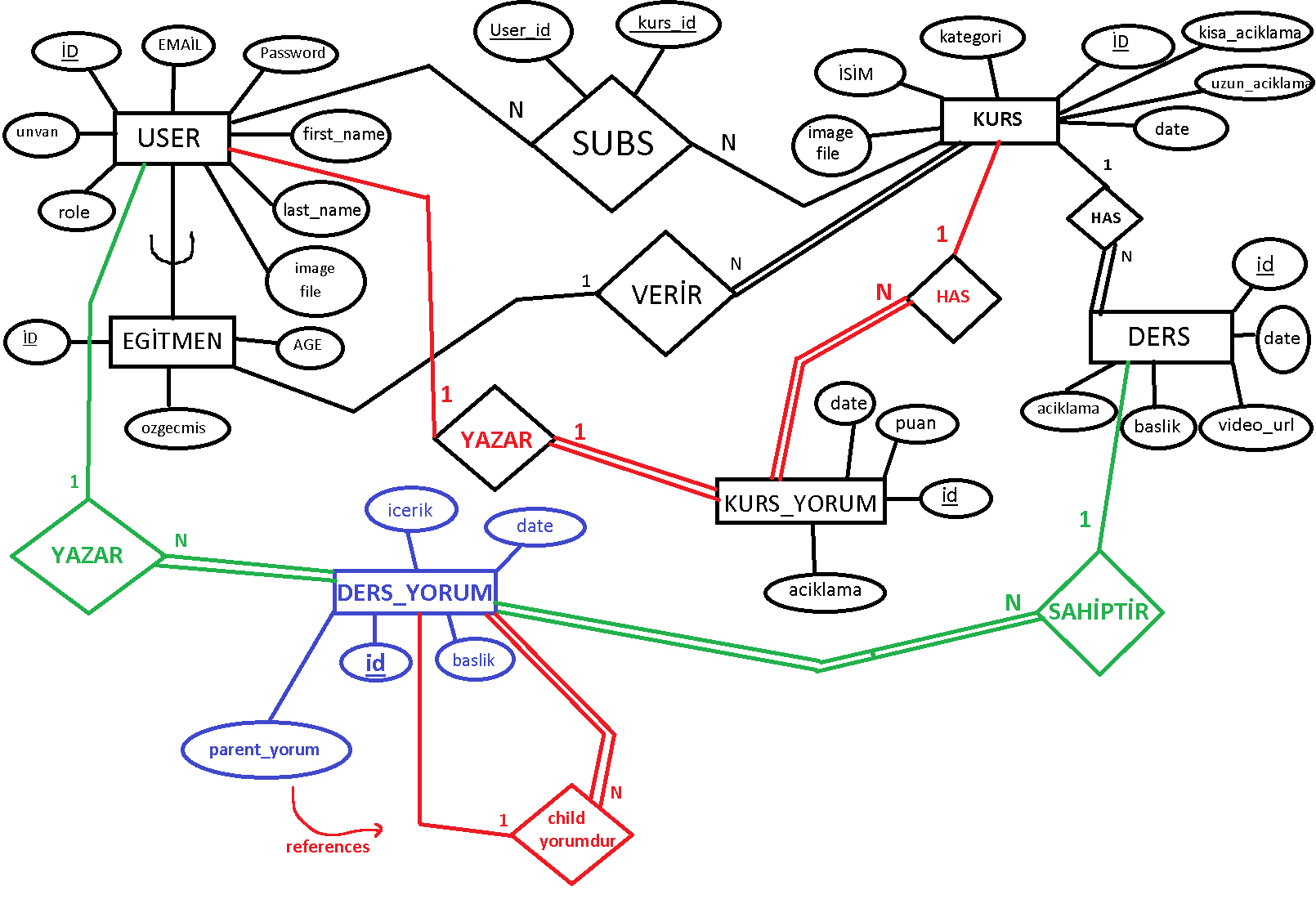
Flask: Python ile yazılmış bir mikro web framework’udur. Belirli tool veya kitaplıklar gerektirmediğinden mikro framework olarak sınıflandırılır. Önceden var olan üçüncü taraf kitaplıklarının ortak işlevler sağladığı diğer bileşenlere sahip değildir. Nesne-ilişkisel eşleyiciler, form doğrulama, karşıya yükleme işleme ve çeşitli authentication teknolojileri mevcuttur.

Bootstrap: Bootstrap da CSS oluşturulan genel kalıplar mevcut. Bu kalıpları bir kere link olarak alınca zor ve uğraştırıcı CSS kısımlarını yazmadan html de class yapısını kullanarak sitenin CSS kısımlarının bir çoğu oluşturulabiliniyor.

Jinja2: Jinja, Python programlama dili için oluşturulmuş bir tasarım motorudur. Pseudo koda benzer ifadeler kullanır. Python dosyaları ve templatesler arasındaki veri aktarımı bu syntax ile gerçekleştirilmiştir. Aynı zamanda Jinja2 html dosyaları arasında kalıtım kolaylığı da sağlar.

SQLite: SQLite, bağımsız, sunucusuz, sıfır yapılandırmalı, transactional bir SQL veritabanı motoru uygulayan in-process bir kitaplıktır. SQLite kodu publictir ve bu nedenle ticari veya özel herhangi bir amaç için kullanımı ücretsizdir. SQLite, embedded bir SQL veritabanı motorudur. Diğer SQL veritabanlarının çoğundan farklı olarak, SQLite ayrı bir sunucu işlemine sahip değildir. SQLite, doğrudan normal disk dosyalarını okur ve yazar. Birden çok tablo, dizin, tetikleyici ve görünüm içeren eksiksiz bir SQL veritabanı, tek bir disk dosyasında bulunur.

**EER DİYAGRAMI**



3NF : Veri Tabanımız baştan düzgün dizayn edildiği için 3NF e uygundur.

User : İD ve e-mail candidate key. Anahtarımız İD. Diğer tüm attributelar ise iD ye fonksiyonel bağımlıdır.

Eğitmen : İD primary key. Age ve özgeçmiş id ye fonksiyonel bağımlıdır.

Kurs : İD ve kurs ismi candidate key. Anahtarımız İD .Diğer tüm attributelar ise iD ye fonksiyonel bağımlıdır.

Ders : İD anahtar. Başlık, içerik, date ve video\_url id ye fonksiyonel bağımlıdır.

Kurs\_yorum : İD anahtar. Puan , date ve açıklama id ye fonksiyonel bağımlıdır.

Ders\_yorum: İD anahtar. başlık , içerik, date ve parent\_yorum id ye fonksiyonel bağımlıdır.

**REFERANSLAR:**

<https://flask.palletsprojects.com/en/2.0.x/>

<https://jinja.palletsprojects.com/en/3.0.x/>

<https://getbootstrap.com>

Sentetik veri oluşturuken ders video url’si için kullandığımız kısıtlanması bulunmayan [YouTube videosu](https://www.youtube.com/watch?v=W7qWa52k-nE&feature=emb_logo)