“

Veritabanı Yönetim Sistemleri

Proje

ÖĞRENCİ ADI	:TARIK
ÖĞRENCİ SOYADI	:BÜKÜCÜ
ÖĞRENCİ NUMARASI	:G191210048
MAİL ADRESİ	:tarik.bukucu@ogr.sakarya.edu.tr
BÖLÜMÜ	:BİLGİSAYAR MÜHENDİSLİĞİ

Uygulamanın Kısa Tanıtımı

Maalesef günümüzün en büyük sorunu Covid-19,bu illete yakalananların durumunu takip eden bir otomasyon dünya sağlığı açısından çok önemli olur.

Covid-19'a yakalanan hastaların durumları takip eden bir otomasyon yapacağım. Hastalık esnasında gösterdikleri belirtiler,hastalığa yakalanan kişilerin yaşları,kullandıkları ilaçlar,hastanede yatanlar,ölenler,kronik rahatsızlık durumu ve bunların hepsini görebilecek olan sağlık çalışanlarının otomasyona giriş tc ve şifresi.Hasta arama,ekleme,silme,durum güncelleme işlemlerni yapan bir otomasyon.

Sonuç olarak, bu hastalığın insanlar üzerinde ilerleyen günlerde ne gibi olumsuzlukları olacağını bilmiyoruz ve bu otomasyon sayesinde olabilecek bir olumsuzluğu erkenden tespit edip belki de bir felaketin önüne geçeceğiz.

İş kuralları

*Programımda kullanıcılar vardır.

*Hasta ve Hekim olmak üzere 2 alt kullanıcı vardır.

*Hekimlerin hasta ekleme,hasta silme,durum güncelleme,hasta arama,hasta değiştirme yetkisi vardır.

*Kullanıcı için TC ve şifre istenir.

*Hastalar ve hekimler kullanıcıdan kalıtım alır.

*Bir hekimin çok adresi olabilir.

*Bir hekimin bir iletişim numarası vardır.

*Bir hastanın çok adresi olabilir.

*Bir hastanın bir iletişim numarası vardır.

*Bir hastanın çok tedavi sonrası yan etkisi olabilir.

*Bir tedavi sonrası yan etki çok hastada görülebilir.

*Bir hastanın çok kronik hastalığı olabilir.

*Bir kronik hastalık çok hastada görülebilir.

*Bir hasta bir kez vefat edebilir.

*Bir hasta çok belirti gösterebilir.

*Bir belirti çok hastada görülebilir.

*Bir hasta çok ilaç kullanabilir.

*Bir ilaç çok hasta tarafından kullanılabilir.

*Bir hasta bir kez yatılı olabilir.

*Bir hasta bir covid aşısı olabilir.

*Bir covid aşısı olan kişide çok yan etki görülebilir.

*Hastaların ve hekimlerin adresleri tutulur.

*Hastaların ve hekimlerin iletişim bilgileri tutulur.

*Covid-19 aşısı olan hastaların bilgileri ve olası yan etkileri tutulur.

*Hastaların tedavi aşamasın yaşadığı hastalık belirtileri tutulur.

*Covid-19 sebebiyle vefat edenlerin listesi tutulur.

*Vefatlar bölümünde pozitif çıktıktan kaç gün sonra vefat ettiği tutulur.

*Covid-19 pozitif vakalardan hastanede tedavi görenlerin listesi tutulur.

*HastanedeYatanların içinde hastanın kaç gün hastanede yattığı ve son durumu tutulur

*Kronik rahatsızlığı olanların rahatsızlıkları tutulur.

*Tedavi sonrası yan etkilere uğrayanların listesi ve yan etkisi.

*tedaviSonrasiYanEtki;hasta,covid-19'u atlattıktan sonra oluşabilecek akciğer hastalıklar vs. etkilerdir.

İlişkisel Şema

kullanici(**tc:int**,password:varchar)

Hasta(**tc:int**,**hastaid:int**,hastaadi:text,hastayasi:int)

HastaAdres(**muhit:varchar**,adresId:int,hastaid:int)

Adres(**adresId:int**,il:varchar,ilce:varchar)

hekimAdres(**muhit:varchar**,adresId:int,hekimId:int)

hekim(hekimAdi:varchar,**hekimId:int,**hekimYasi:int,**tc:int**)

hastailetisim(hastaid:int,iletisimid:int,**telefonNo:int**)

hekimiletisim(hekimid:int,iletisimid:int,**telefonNo:int**)

beklenmedikHastalik(**hastaid:int,sonYanEtkiId:int**)

kronikHastalik(hastaid:int,**hastalikid:int**,hastalikSayisi:int)

tedaviSonrasiYanEtki(hastaid:int,rahatsizlikSebebi:varchar,sonDurum:varchar,**sonYanEtkiId:int**)

kritikDurum(**hastaid:int,kronikHastalikId:int**)

vefatlar(hastaid:int,vefatGünü:int,**vefatSiraNo:int**)

hastaBelirti(**belirtiId:int,hastaId:int**)

belirtiler(**belirtiId:int**,belirtiler:varchar,belirtiSayisi:int,hastaid:int)

hastailac(**hastaid:int,ilacId:int**)

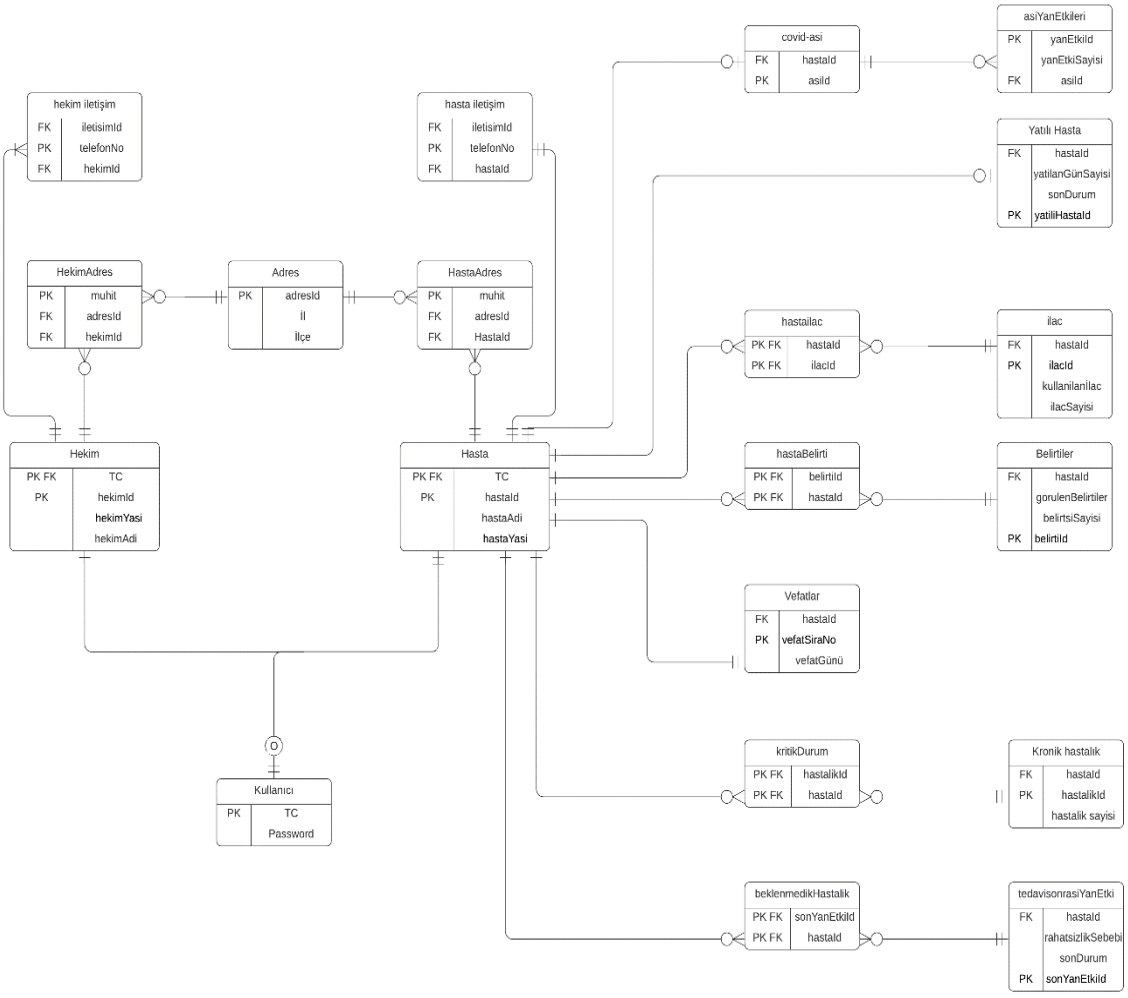
ilac(hastaid:int,**ilacId:int**,ilacSayisi:int,kullanilanilac:varchar)

yatiliHasta(hastaid:int,sonDurum:varchar,yatilanGunSayisi:int,**yatiliHastaId:int**)

covidAsi(**asiId:int**,hastaid:int)

asiYanEtki(asiId:int,**yanEtkiId:int**,yanEtkiSayisi:int)

Varlık Bağıntı Modeli



SQL İfadeler

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 13.1
-- Dumped by pg_dump version 13.1

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: adres; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.adres (
    "adresId" integer NOT NULL,
```

```
    il character varying NOT NULL,
    ilce character varying NOT NULL
);


ALTER TABLE public.adres OWNER TO postgres;

--
-- Name: asiYanEtki; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."asiYanEtki" (
    "yanEtkiId" integer NOT NULL,
    "yanEtkiSayisi" integer NOT NULL,
    "asiId" integer NOT NULL
);


ALTER TABLE public."asiYanEtki" OWNER TO postgres;

--
-- Name: beklenmedikHastalik; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."beklenmedikHastalik" (
    "sonYanEtkiId" integer NOT NULL,
    "hastaId" integer NOT NULL
);
```

```
ALTER TABLE public."beklenmedikHastalik" OWNER TO postgres;

--
-- Name: belirtiler; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.belirtiler (
    "hastaId" integer NOT NULL,
    "belirtiSayisi" integer NOT NULL,
    belirtiler character varying NOT NULL,
    "belirtiId" integer NOT NULL
);


ALTER TABLE public.belirtiler OWNER TO postgres;

--
-- Name: covidAsi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."covidAsi" (
    "asiId" integer NOT NULL,
    "hastaId" integer NOT NULL
);


ALTER TABLE public."covidAsi" OWNER TO postgres;

--
-- Name: hasta; Type: TABLE; Schema: public; Owner: postgres
```

```
--

CREATE TABLE public.hasta (
    hastaid integer NOT NULL,
    hastaadi text NOT NULL,
    hastayasi integer NOT NULL,
    tc text NOT NULL
);


ALTER TABLE public.hasta OWNER TO postgres;

--
-- Name: hastaAdres; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."hastaAdres" (
    muhit character varying NOT NULL,
    "adresId" integer NOT NULL,
    "hastaId" integer NOT NULL
);


ALTER TABLE public."hastaAdres" OWNER TO postgres;

--
-- Name: hastaBelirti; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."hastaBelirti" (
```

```
    "belirtiId" integer NOT NULL,
    "hastaId" integer NOT NULL
);


ALTER TABLE public."hastaBelirti" OWNER TO postgres;

--
-- Name: hastaIletisim; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."hastaIletisim" (
    "iletisimId" integer NOT NULL,
    "hastaId" integer NOT NULL,
    "telefonNo" integer NOT NULL
);


ALTER TABLE public."hastaIletisim" OWNER TO postgres;

--
-- Name: hastailac; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.hastailac (
    "hastaId" integer NOT NULL,
    "ilacId" integer NOT NULL
);
```

```
ALTER TABLE public.hastailac OWNER TO postgres;

--
-- Name: hekim; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.hekim (
    "hekimId" integer NOT NULL,
    "hekimAdi" character varying NOT NULL,
    "hekimYasi" integer NOT NULL,
    "TC" integer NOT NULL
);


ALTER TABLE public.hekim OWNER TO postgres;

--
-- Name: hekimAdres; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."hekimAdres" (
    muhit character varying NOT NULL,
    "adresId" integer NOT NULL,
    "hekimId" integer NOT NULL
);


ALTER TABLE public."hekimAdres" OWNER TO postgres;

--
```

```
-- Name: hekimIletisim; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."hekimIletisim" (
    "iletisimId" integer NOT NULL,
    "hekimId" integer NOT NULL,
    "telefonNo" integer NOT NULL
);


ALTER TABLE public."hekimIletisim" OWNER TO postgres;

--
-- Name: ilac; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.ilac (
    "ilacId" integer NOT NULL,
    "hastaId" integer NOT NULL,
    "kullanilanIlac" character varying NOT NULL,
    "ilacSayisi" integer NOT NULL
);


ALTER TABLE public.ilac OWNER TO postgres;

--
-- Name: kritikDurum; Type: TABLE; Schema: public; Owner: postgres
--
```

```
CREATE TABLE public."kritikDurum" (
    "hastaId" integer NOT NULL,
    "kronikHastalikId" integer NOT NULL
);


ALTER TABLE public."kritikDurum" OWNER TO postgres;

--
-- Name: kronikHastalik; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."kronikHastalik" (
    "hastaId" integer NOT NULL,
    "hastalikId" integer NOT NULL,
    "hastalikSayisi" integer NOT NULL,
    "kronikHastalikId" integer NOT NULL
);


ALTER TABLE public."kronikHastalik" OWNER TO postgres;

--
-- Name: kullanici; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.kullanici (
    tc integer NOT NULL,
    password character varying NOT NULL
);
```

```
ALTER TABLE public.kullanici OWNER TO postgres;

--
-- Name: tedaviSonrasiYanEtki; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."tedaviSonrasiYanEtki" (
    "hastaId" integer NOT NULL,
    "rahatsizlikSebebi" character varying NOT NULL,
    "sonDurum" character varying NOT NULL,
    "sonYanEtkiId" integer NOT NULL
);


ALTER TABLE public."tedaviSonrasiYanEtki" OWNER TO postgres;

--
-- Name: vefatlar; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.vefatlar (
    "hastaId" integer NOT NULL,
    "vefatGunu" integer NOT NULL,
    "vefatSiraNo" integer NOT NULL
);


ALTER TABLE public.vefatlar OWNER TO postgres;
```

```
--
-- Name: yatiliHasta; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."yatiliHasta" (
    "hastaId" integer NOT NULL,
    "yatilanGünSayisi" integer NOT NULL,
    "sonDurum" character varying NOT NULL,
    "yatiliHastaId" integer NOT NULL
);


ALTER TABLE public."yatiliHasta" OWNER TO postgres;

--
-- Data for Name: adres; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public.adres VALUES
        (60000, 'sakarya', 'serdivan');


--
-- Data for Name: asiYanEtki; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
```

```
-- Data for Name: beklenmedikHastalik; Type: TABLE DATA; Schema: public; Owner:
postgres
--

INSERT INTO public."beklenmedikHastalik" VALUES
        (40009, 30009);



--
-- Data for Name: belirtiler; Type: TABLE DATA; Schema: public; Owner: postgres
--




--
-- Data for Name: covidAsi; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."covidAsi" VALUES
        (50000, 30009);



--
-- Data for Name: hasta; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public.hasta VALUES
        (30000, 'Aziz', 33, '10000'),
        (30001, 'Sancar', 55, '10001'),
        (30002, 'Ahmet', 47, '10002'),
        (30003, 'Mehmet', 39, '10003'),
```

```
	(30004, 'Ayşe', 40, '10004'),
	(30005, 'Beyza', 21, '10005'),
	(30006, 'Nisa', 22, '10006'),
	(30007, 'Cansu', 23, '10007'),
	(30008, 'Fatma', 27, '10008'),
	(30009, 'Bahar', 33, '10009');


--
-- Data for Name: hastaAdres; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."hastaAdres" VALUES
	('no3', 60000, 30000);


--
-- Data for Name: hastaBelirti; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."hastaBelirti" VALUES
	(70008, 30008),
	(70005, 30005);


--
-- Data for Name: hastaIletisim; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."hastaIletisim" VALUES
```

```
        (80005, 30005, 50577744);


--
-- Data for Name: hastailac; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public.hastailac VALUES
        (30005, 90005);


--
-- Data for Name: hekim; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public.hekim VALUES
        (20010, 'TARIK', 22, 10010);


--
-- Data for Name: hekimAdres; Type: TABLE DATA; Schema: public; Owner: postgres
--


--
-- Data for Name: hekimIletisim; Type: TABLE DATA; Schema: public; Owner: postgres
--
```

```
--
-- Data for Name: ilac; Type: TABLE DATA; Schema: public; Owner: postgres
--




--
-- Data for Name: kritikDurum; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."kritikDurum" VALUES
        (30002, 100002);



--
-- Data for Name: kronikHastalik; Type: TABLE DATA; Schema: public; Owner: postgres
--




--
-- Data for Name: kullanici; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public.kullanici VALUES
        (10000, '1234'),
        (10001, '1234'),
        (10002, '1234'),
        (10003, '1234'),
```

(10004, '1234'),

(10005, '1234'),

(10006, '1234'),

(10007, '1234'),

(10008, '1234'),

(10009, '1234'),

(10010, '1234');

--

-- Data for Name: tedaviSonrasiYanEtki; Type: TABLE DATA; Schema: public; Owner: postgres

--

--

-- Data for Name: vefatlar; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public.vefatlar VALUES

(30001, 37, 200001);

--

-- Data for Name: yatiliHasta; Type: TABLE DATA; Schema: public; Owner: postgres

--

INSERT INTO public."yatiliHasta" VALUES

(30001, 27, 'saglikli', 300001);

--

-- Name: adres adres_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.adres
    ADD CONSTRAINT adres_pkey PRIMARY KEY ("adresId");
```

--

-- Name: asiYanEtki asiYanEtki_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public."asiYanEtki"
    ADD CONSTRAINT "asiYanEtki_pkey" PRIMARY KEY ("yanEtkiId");
```

--

-- Name: covidAsi asi_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public."covidAsi"
    ADD CONSTRAINT asi_pkey PRIMARY KEY ("asiId");
```

--

-- Name: beklenmedikHastalik beklenmedikHastalik_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public."beklenmedikHastalik"
    ADD CONSTRAINT "beklenmedikHastalik_pkey" PRIMARY KEY ("sonYanEtkiId",
"hastaId");



--
-- Name: belirtiler belirtiler_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.belirtiler
    ADD CONSTRAINT belirtiler_pkey PRIMARY KEY ("belirtiId");



--
-- Name: hastaAdres hastaAdres_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."hastaAdres"
    ADD CONSTRAINT "hastaAdres_pkey" PRIMARY KEY (muhit);



--
-- Name: hastaBelirti hastaBelirti_pkey; Type: CONSTRAINT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."hastaBelirti"
    ADD CONSTRAINT "hastaBelirti_pkey" PRIMARY KEY ("belirtiId", "hastaId");



--
```

```
-- Name: hastaIletisim hastaIletisim_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."hastaIletisim"
    ADD CONSTRAINT "hastaIletisim_pkey" PRIMARY KEY ("telefonNo");



--
-- Name: hasta hasta_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.hasta
    ADD CONSTRAINT hasta_pkey PRIMARY KEY (hastaid);



--
-- Name: hastailac hastailac_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.hastailac
    ADD CONSTRAINT hastailac_pkey PRIMARY KEY ("hastaId", "ilacId");



--
-- Name: hekimAdres hekimAdres_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."hekimAdres"
    ADD CONSTRAINT "hekimAdres_pkey" PRIMARY KEY (muhit);
```

--

-- Name: hekimIletisim hekimIletisim_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hekimIletisim"

    ADD CONSTRAINT "hekimIletisim_pkey" PRIMARY KEY ("telefonNo");

--

-- Name: hekim hekim_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hekim

    ADD CONSTRAINT hekim_pkey PRIMARY KEY ("hekimId", "TC");

--

-- Name: kritikDurum kritikDurum_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."kritikDurum"

    ADD CONSTRAINT "kritikDurum_pkey" PRIMARY KEY ("hastaId", "kronikHastalikId");

--

-- Name: kronikHastalik kronikHastalik_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."kronikHastalik"

    ADD CONSTRAINT "kronikHastalik_pkey" PRIMARY KEY ("hastalikId", "kronikHastalikId");

--

-- Name: kullanici kullanici_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.kullanici

    ADD CONSTRAINT kullanici_pkey PRIMARY KEY (tc);

--

-- Name: ilac kullanilanIlac_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ilac

    ADD CONSTRAINT "kullanilanIlac_pkey" PRIMARY KEY ("ilacId");

--

-- Name: tedaviSonrasiYanEtki tedaviSonrasiYanEtki_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."tedaviSonrasiYanEtki"

    ADD CONSTRAINT "tedaviSonrasiYanEtki_pkey" PRIMARY KEY ("sonYanEtkiId");

--

-- Name: beklenmedikHastalik unique_beklenmedikHastalik_hastaId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public."beklenmedikHastalik"
    ADD CONSTRAINT "unique_beklenmedikHastalik_hastaId" UNIQUE ("hastaId");
```

--

-- Name: beklenmedikHastalik unique_beklenmedikHastalik_sonYanEtkiId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public."beklenmedikHastalik"
    ADD CONSTRAINT "unique_beklenmedikHastalik_sonYanEtkiId" UNIQUE ("sonYanEtkiId");
```

--

-- Name: belirtiler unique_belirtiler_hastaId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.belirtiler
    ADD CONSTRAINT "unique_belirtiler_hastaId" UNIQUE ("hastaId");
```

--

-- Name: hastaBelirti unique_hastaBelirti_belirtiId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hastaBelirti"

    ADD CONSTRAINT "unique_hastaBelirti_belirtiId" UNIQUE ("belirtiId");

--

-- Name: hastaBelirti unique_hastaBelirti_hastaId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hastaBelirti"

    ADD CONSTRAINT "unique_hastaBelirti_hastaId" UNIQUE ("hastaId");

--

-- Name: hasta unique_hasta_hastaId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hasta

    ADD CONSTRAINT "unique_hasta_hastaId" UNIQUE (hastaid);

--

-- Name: hasta unique_hasta_tc; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hasta

    ADD CONSTRAINT unique_hasta_tc UNIQUE (tc);

--

-- Name: hastailac unique_hastailac_hastaId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hastailac

    ADD CONSTRAINT "unique_hastailac_hastaId" UNIQUE ("hastaId");

--

-- Name: hastailac unique_hastailac_ilacId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hastailac

    ADD CONSTRAINT "unique_hastailac_ilacId" UNIQUE ("ilacId");

--

-- Name: hekim unique_hekim_TC; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hekim

    ADD CONSTRAINT "unique_hekim_TC" UNIQUE ("TC");

--

-- Name: hekim unique_hekim_hekimId; Type: CONSTRAINT; Schema: public; Owner: postgres

--

```
ALTER TABLE ONLY public.hekim
    ADD CONSTRAINT "unique_hekim_hekimId" UNIQUE ("hekimId");


--
-- Name: kritikDurum unique_kritikDurum_kronikHastalikId; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."kritikDurum"
    ADD CONSTRAINT "unique_kritikDurum_kronikHastalikId" UNIQUE ("kronikHastalikId");


--
-- Name: kronikHastalik unique_kronikHastalik_kronikHastalikId; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."kronikHastalik"
    ADD CONSTRAINT "unique_kronikHastalik_kronikHastalikId" UNIQUE ("kronikHastalikId");


--
-- Name: vefatlar vefatlar_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.vefatlar
    ADD CONSTRAINT vefatlar_pkey PRIMARY KEY ("vefatSiraNo");
```

--

-- Name: yatiliHasta yatiliHasta_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."yatiliHasta"

    ADD CONSTRAINT "yatiliHasta_pkey" PRIMARY KEY ("yatiliHastaId");

--

-- Name: hastaAdres adres-hastaAdres; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hastaAdres"

    ADD CONSTRAINT "adres-hastaAdres" FOREIGN KEY ("adresId") REFERENCES public.adres("adresId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hekimAdres adres-hekimAdres; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hekimAdres"

    ADD CONSTRAINT "adres-hekimAdres" FOREIGN KEY ("adresId") REFERENCES public.adres("adresId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: tedaviSonrasiYanEtki beklenmedikHastalik-tedaviSonrasiYanEtki; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."tedaviSonrasiYanEtki"

ADD CONSTRAINT "beklenmedikHastalik-tedaviSonrasiYanEtki" FOREIGN KEY ("sonYanEtkiId") REFERENCES public."beklenmedikHastalik"("sonYanEtkiId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: asiYanEtki covidAsi-asiYanEtki; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."asiYanEtki"

ADD CONSTRAINT "covidAsi-asiYanEtki" FOREIGN KEY ("asiId") REFERENCES public."covidAsi"("asiId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: beklenmedikHastalik hasta-beklenmedikHastalik; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."beklenmedikHastalik"

ADD CONSTRAINT "hasta-beklenmedikHastalik" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: covidAsi hasta-covidAsi; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."covidAsi"

ADD CONSTRAINT "hasta-covidAsi" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hastaAdres hasta-hastaAdres; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hastaAdres"

ADD CONSTRAINT "hasta-hastaAdres" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hastaBelirti hasta-hastaBelirti; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hastaBelirti"

ADD CONSTRAINT "hasta-hastaBelirti" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hastailac hasta-hastailac; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.hastailac

ADD CONSTRAINT "hasta-hastailac" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hastaIletisim hasta-hastailetisim; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hastaIletisim"

    ADD CONSTRAINT "hasta-hastailetisim" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: kritikDurum hasta-kritikDurum; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."kritikDurum"

    ADD CONSTRAINT "hasta-kritikDurum" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: vefatlar hasta-vefat; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.vefatlar

    ADD CONSTRAINT "hasta-vefat" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: yatiliHasta hasta-yatiliHasta; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."yatiliHasta"

ADD CONSTRAINT "hasta-yatiliHasta" FOREIGN KEY ("hastaId") REFERENCES public.hasta(hastaid) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: belirtiler hastaBelirti-belirtiler; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.belirtiler

ADD CONSTRAINT "hastaBelirti-belirtiler" FOREIGN KEY ("belirtiId") REFERENCES public."hastaBelirti"("belirtiId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: ilac hastailac-ilac; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public.ilac

ADD CONSTRAINT "hastailac-ilac" FOREIGN KEY ("ilacId") REFERENCES public.hastailac("ilacId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hekimAdres hekim-hekimAdres; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hekimAdres"

ADD CONSTRAINT "hekim-hekimAdres" FOREIGN KEY ("hekimId") REFERENCES public.hekim("hekimId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hekimIletisim hekim-hekimiletisim; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."hekimIletisim"

ADD CONSTRAINT "hekim-hekimiletisim" FOREIGN KEY ("hekimId") REFERENCES public.hekim("hekimId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: kronikHastalik kritikDurum-kronikHastalik; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."kronikHastalik"

ADD CONSTRAINT "kritikDurum-kronikHastalik" FOREIGN KEY ("kronikHastalikId") REFERENCES public."kritikDurum"("kronikHastalikId") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

--

-- Name: hekim kullanici-hekim; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

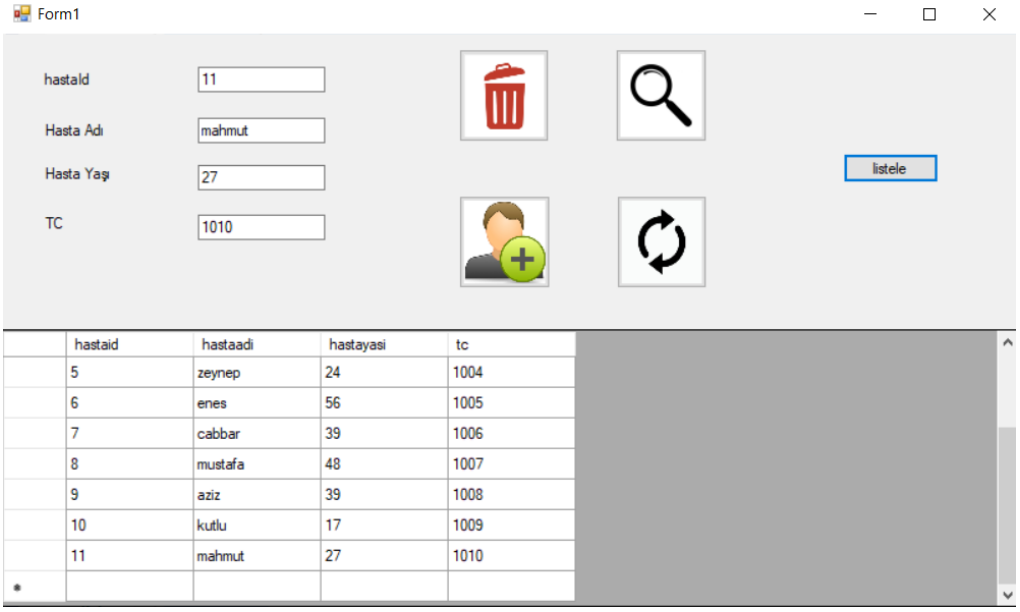
```
ALTER TABLE ONLY public.hekim
    ADD CONSTRAINT "kullanici-hekim" FOREIGN KEY ("TC") REFERENCES public.kullanici(tc) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;



--
-- PostgreSQL database dump complete
--
```

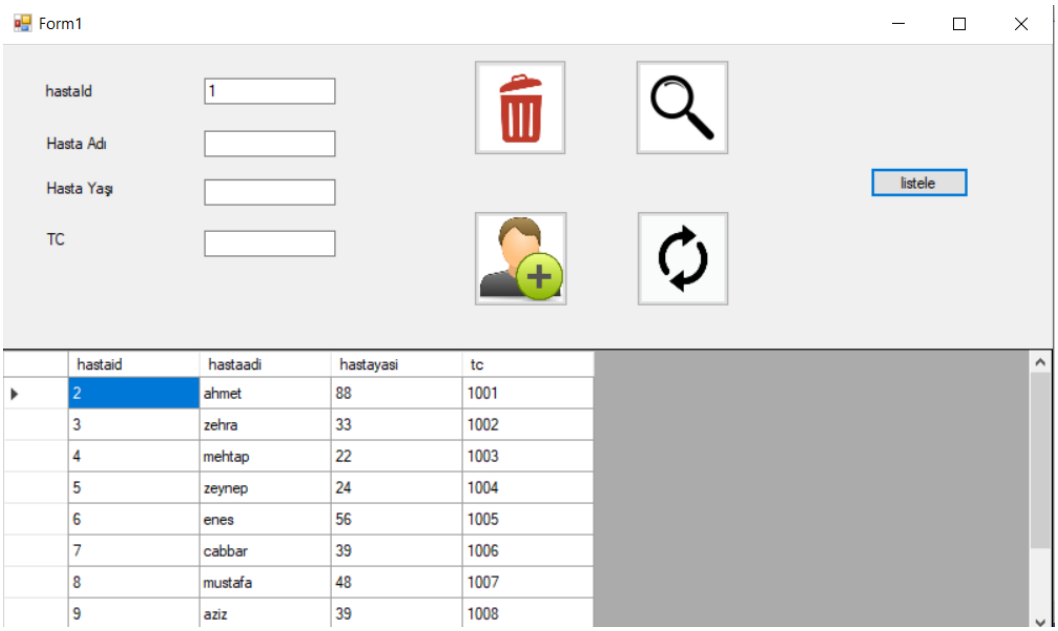
Komutlara ait Ekran Görüntüleri

Ekle Komutu:11 hastaid'li hastamız Mahmut Bey'in yaşı ve tc kimlik numarası sisteme girilip hasta ekle simgesine tıkladığımızda hastamız listenin sonuna eklenir.



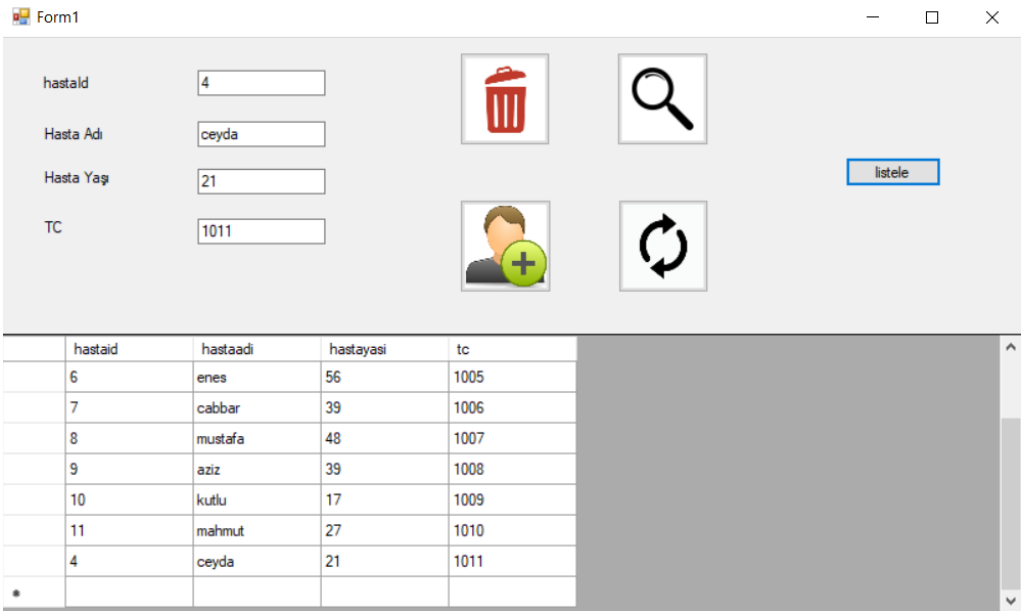
hastaid	hastaadi	hastayasi	tc
5	zeynep	24	1004
6	enes	56	1005
7	cabbar	39	1006
8	mustafa	48	1007
9	aziz	39	1008
10	kutlu	17	1009
11	mahmut	27	1010

Sil Komutu:hastaid değeri girilerek ve çöp kutusuna tıklanarak listeden silinmesi istenilen 1 hastaid'li hastamız listeden silinir.

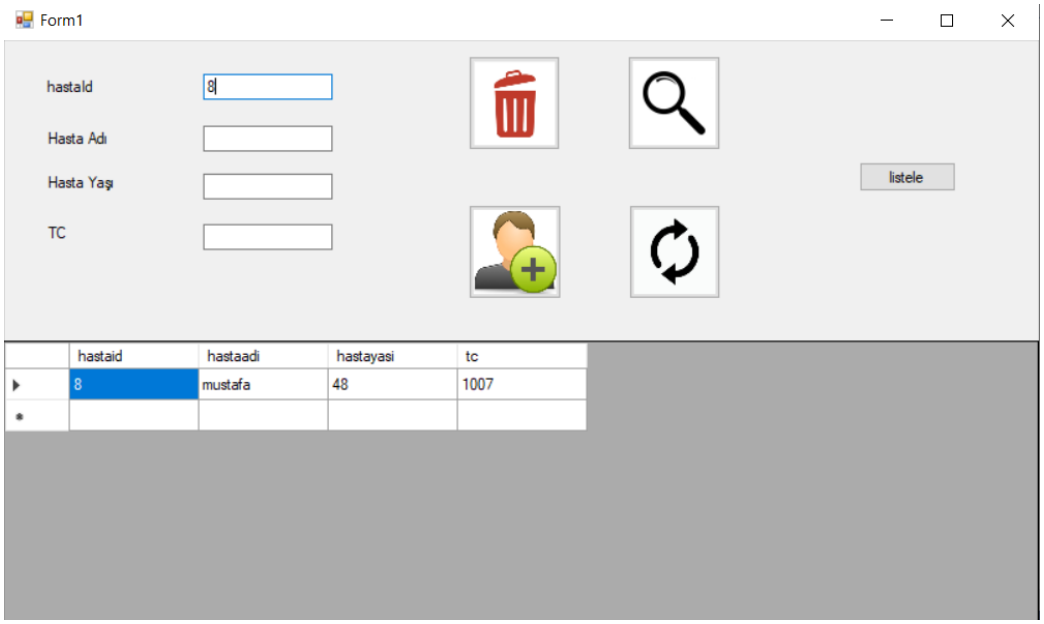


hastaid	hastaadi	hastayasi	tc
2	ahmet	88	1001
3	zehra	33	1002
4	mehtap	22	1003
5	zeynep	24	1004
6	enes	56	1005
7	cabbar	39	1006
8	mustafa	48	1007
9	aziz	39	1008

Güncelle Komutu:4 numaralı hastamız Mehtap Hanım yerine hastaid kısmına 4 yazıp yerine gelecek hastanın bilgilerini yazıp güncelle butonuna tıkladığımızda yerine Ceyda Hanım geliyor.



Bul Komutu: Aramak istediğimiz hastanın tc numarasını ya da hastaid numarasını girip find butonuna bastığımızda bulmak istediğimiz hastanın bilgileri karşımıza çıkar.



Login ekranımın ekran görüntüsü;

Hasta listemin ilk hali;

Database: yedek covid19 Schema: public Table

	hastaid	hastaadi	hastayasi	tc
1	1	mehmet	78	1000
2	2	ahmet	88	1001
3	3	zehra	33	1002
4	4	mehtap	22	1003
5	5	zeynep	24	1004
6	6	enes	56	1005
7	7	cabbar	39	1006
8	8	mustafa	48	1007
9	9	aziz	39	1008
10	10	kutlu	17	1009

Hasta listemin ekle,sil,güncelle,bul komutlarını 1'er kez kullandıktan sonraki hali;

Database: yedek covid19 Schema: public Tabl

	hastaid	hastaadi	hastayasi	tc
1	2	ahmet	88	1001
2	3	zehra	33	1002
3	4	ceyda	21	1011
4	5	zeynep	24	1004
5	6	enes	56	1005
6	7	cabbar	39	1006
7	8	mustafa	48	1007
8	9	aziz	39	1008
9	10	kutlu	17	1009
10	11	mahmut	27	1010

Uygulamanın Kaynak Kodları(C#)

FORM1.CS(UYGULAMA EKRANI)

```
using Npgsql;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;


namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port=5432;
Database=yedek covid19; user ID=postgres ; password=1234");



        //Hasta Listeleme
        private void button4_Click(object sender, EventArgs e)
        {
            yenile();
        }


        //Hasta silme

        private void button3_Click(object sender, EventArgs e)
        {
            baglanti.Open();

            NpgsqlCommand komut2 = new NpgsqlCommand("Delete from hasta where
hastaid=@p1",baglanti);
            komut2.Parameters.AddWithValue("@p1", int.Parse(textBox1.Text));
            komut2.ExecuteNonQuery();

            baglanti.Close();
            yenile();
        }


        //Hasta ekleme

        private void button1_Click(object sender, EventArgs e)
        {
            baglanti.Open();
```

```
            NpgsqlCommand komut1 = new NpgsqlCommand("insert into hasta
(hastaid,hastaadi,hastayasi,tc) values (@p1,@p2,@p3,@p4)", baglanti);
            komut1.Parameters.AddWithValue("@p1", int.Parse(textBox1.Text));
            komut1.Parameters.AddWithValue("@p2", textBox2.Text);
            komut1.Parameters.AddWithValue("@p3", int.Parse(textBox3.Text));
            komut1.Parameters.AddWithValue("@p4", int.Parse(textBox4.Text));
            komut1.ExecuteNonQuery();

            baglanti.Close();
            yenile();

        }



        //Hasta arama

        private void button6_Click(object sender, EventArgs e)
        {
            baglanti.Open();

            DataTable tablo = new DataTable();
            NpgsqlDataAdapter adtr = new NpgsqlDataAdapter("select * from hasta where
tc like'%" + textBox4.Text + "%'", baglanti);
            adtr.Fill(tablo);
            dataGridView1.DataSource = tablo;
            baglanti.Close();
        }



        //Hasta Guncelle

        private void button2_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            NpgsqlCommand komut3 = new NpgsqlCommand("update hasta set
hastaadi=@p2,hastayasi=@p3,tc=@p4 where hastaid=@p1", baglanti);

            komut3.Parameters.AddWithValue("@p1", int.Parse(textBox1.Text));
            komut3.Parameters.AddWithValue("@p2", textBox2.Text);
            komut3.Parameters.AddWithValue("@p3", int.Parse(textBox3.Text));
            komut3.Parameters.AddWithValue("@p4", int.Parse(textBox4.Text));
            komut3.ExecuteNonQuery();

            baglanti.Close();
            yenile();
        }

        public int yenile()
        {
            string sorgu = "select * from hasta ";
            NpgsqlDataAdapter da = new NpgsqlDataAdapter(sorgu, baglanti);
            DataSet ds = new DataSet();
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
            return 0;

        }
    }
}
```

FORM2.CS(LOGIN EKRANI)

```
using Npgsql;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }
        NpgsqlConnection baglanti = new NpgsqlConnection("server=localHost; port=5432;
Database=yedek covid19; user ID=postgres ; password=1234");



        private void label1_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            baglanti.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {

            baglanti.Open();
            NpgsqlDataAdapter da = new NpgsqlDataAdapter("select *from kullanici where
tc ='" + int.Parse(textBox1.Text) + "' and password='" + textBox2.Text + "'",
baglanti);
            MessageBox.Show("Giriş başarıyla gerçekleşti.", "Giriş yapıldı.",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
            DataSet ds = new DataSet();
            da.Fill(ds);
            this.Hide();
            Form1 fm = new Form1();
            fm.ShowDialog();

            baglanti.Close();

        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }
    }
 }
```

Projemi anlattığım video linki

youtube:TARIK Bkc

https://youtu.be/KyAi4Zg-f9A