

# E-TİCARET SİTESİ

WEB PROGRAMLAMA  
PROJE RAPORU

*İDRİS BÖLGE - 162113034 – N.Ö(2.GRUP)*

*FERHAT BAŞARAN - 172114211 – İ.Ö(2.GRUP)*

## VERİTABANI :

*Basic\_app.sqlite*

İsim	Tip
▼ Tablolar (7)	
▼ GecmisSepetler	
id	INTEGER
urunadi	VARCHAR ( 140 )
urunAdet	VARCHAR ( 20 )
timestamp	DATETIME
user_id	INTEGER
▼ roles	
id	INTEGER
name	VARCHAR ( 50 )
▼ sepetler	
id	INTEGER
urunadi	VARCHAR ( 140 )
urunAdet	varchar ( 140 )
timestamp	DATETIME
user_id	INTEGER
▼ user_roles	
id	INTEGER
user_id	INTEGER
role_id	INTEGER
users	

## Data.db

▼ Tablolar (3)	
▼ Kullanici	
Id	INTEGER
KullaniciAdi	TEXT
Sifre	TEXT
Yetki	INTEGER
▼ Urunler	
id	INTEGER
urunadi	TEXT
urunfiyat	INTEGER

# Ekran Görüntüleri

## Anasayfa

### E- Ticaret Sitesi

[Sign in](#)[Home page](#)

#### Ana sayfa

#### Bu web sitesinin anasayfasıdır

[Kayıt ol](#)[Giriş Yap](#)[Ana sayfa \(Herkes erişebilir\)](#)[Üye sayfası \(Giriş gerekli: member@example.com / Password1\)](#)[Admin sayfası \(Rol gerekli: admin@example.com / Password1\)](#)[Güncellenen ürünler](#)[Urun Satış Bilgisi](#)[Urun Satış Bilgisi\(Tarihli\)](#)[Çıkış](#)

```
# Sayfanın Giriş sayfası
@app.route('/')
def home_page():
    return render_template_string("""
        {% extends "flask_user_layout.html" %}
        {% block content %}
            <h2>{%trans%}Ana sayfa{%endtrans%}</h2>
            <h1>{%trans%}Bu web sitesinin anasayfasıdır{%endtrans%}</h1>
            <p><a href={{ url_for('user.register') }}>{%trans%}Kayıt ol{%endtrans%}</a></p>
            <p><a href={{ url_for('user.login') }}>{%trans%}Giriş Yap{%endtrans%}</a></p>
            <p><a href={{ url_for('home_page') }}>{%trans%}Ana sayfa{%endtrans%}</a> (Herkes erişebilir)</p>
            <p><a href={{ url_for('member_page') }}>{%trans%}Üye sayfası{%endtrans%}</a> (Giriş gerekli: member@example.com / Password1)</p>
            <p><a href={{ url_for('admin_page') }}>{%trans%}Admin sayfası{%endtrans%}</a> (Rol gerekli: admin@example.com / Password1)</p>
            <p><a href={{ url_for('guncellemeler') }}>{%trans%}Güncellenen ürünler{%endtrans%}</a></p>
            <p><a href={{ url_for('urunsatisbilgisi') }}>{%trans%}Urun Satış Bilgisi{%endtrans%}</a></p>
            <p><a href={{ url_for('urunsatisbilgisitarih') }}>{%trans%}Urun Satış Bilgisi(Tarihli){%endtrans%}</a></p>

            <p><a href={{ url_for('user.logout') }}>{%trans%}Çıkış{%endtrans%}</a></p>
        {% endblock %}
    """)
```

## Üyeler

### Üyeler sayfası

## Sisteme giriş yapanlar bu sayfayı görüntüleyebilir

[Gönderi İşlemleri](#)

[Sepet İşlemleri](#)

[Kayıt ol](#)

[Giriş Yap](#)

[Ana sayfa](#) (Herkes erişebilir)

[Üye sayfası](#) (Giriş gerekli: member@example.com / Password1)

[Admin sayfası](#) (Rol gerekli: admin@example.com / Password1)

[Çıkış](#)

```
# Üyelerin Sayfası
@app.route('/members')
@login_required
# @login_required # Use of @login_required decorator
def member_page():
    return render_template_string("""
        {% extends "flask_user_layout.html" %}
        {% block content %}
        <h2>{{trans}}Üyeler sayfası {{endtrans}}</h2>
        <h1>{{trans}} Sisteme giriş yapanlar bu sayfayı görüntüleyebilir{{endtrans}}</h1>
        <p><a href={{ url_for('gonderi_islemleri') }}>{{trans}}Gönderi İşlemleri{{endtrans}}</a></p>
        <p><a href={{ url_for('Sepet_islemleri') }}>{{trans}}Sepet İşlemleri{{endtrans}}</a></p>
        <p><a href={{ url_for('user.register') }}>{{trans}}Kayıt ol{{endtrans}}</a></p>
        <p><a href={{ url_for('user.login') }}>{{trans}}Giriş Yap{{endtrans}}</a></p>
        <p><a href={{ url_for('home_page') }}>{{trans}}Ana sayfa{{endtrans}}</a> (Herkes erişebilir)</p>
        <p><a href={{ url_for('member_page') }}>{{trans}}Üye sayfası{{endtrans}}</a> (Giriş gerekli: member@example.com / Password1)</p>
        <p><a href={{ url_for('admin_page') }}>{{trans}}Admin sayfası{{endtrans}}</a> (Rol gerekli: admin@example.com / Password1)</p>
        <p><a href={{ url_for('user.logout') }}>{{trans}}Çıkış{{endtrans}}</a></p>
        {% endblock %}
    """)
```

## Sepet İşlemleri

[Ürünleri Listele](#)

[Ürün Satın Al](#)

[Alışveriş Sepetin](#)

[Geçmiş Alışveriş Listesi](#)

### Ürünleri Listele

Ürün Adı	Ürün Fiyatı
Uçlu kalem	7
kağıt	2
silgi	10
su	1
asda	7
Bardak	20
Kitap	24
Çanta	100
Bilgisayar	2000

[Anasayfaya dön](#)

```
@app.route('/urunlistesi')
def urunlistesi():
    con = sqlite3.connect("Data.db")
    con.row_factory = sqlite3.Row

    cur = con.cursor()
    cur.execute("select * from Urunler")

    rows = cur.fetchall()
    return render_template("Urunler.html", rows = rows)

#Ürün satın alma sayfasına yönlendirme
@app.route('/urunsatinal')
def urunsatinal():
    con = sqlite3.connect("Data.db")
    con.row_factory = sqlite3.Row
    cur = con.cursor()
    cur.execute("select * from Urunler")
    rows = cur.fetchall()
    return render_template("UrunSatin.html", rows = rows)
```

## Ürün Satın Al

Ürün Adı	Ürün Fiyatı
Uçlu kalem	7
kağıt	2
silgi	10
su	1
asda	7
Bardak	20
Kitap	24
Çanta	100
Bilgisayar	2000

```
@app.route('/urunsatinal')
def urunsatinal():
    con = sqlite3.connect("Data.db")
    con.row_factory = sqlite3.Row
    cur = con.cursor()
    cur.execute("select * from Urunler")
    rows = cur.fetchall()
    return render_template("UrunSatin.html", rows = rows)
```

## Üye Sayfası

## Ürün Satın

[Anasayfaya dön](#)

```
@app.route('/SepeteEkle', methods = ['POST', 'GET'])
@login_required
def SepeteEkle():
    if request.method == 'POST':
        try:
            urunadi = request.form['urunadi']
            urunAdet = request.form['urunAdet']
            sepet = Sepet(urunadi = urunadi, urunAdet = urunAdet, user_id = current_user.id)
            db.session.add(sepet)
            db.session.commit()
            msg = "Sepete başarı ile eklendi"
        except:
            msg = "Sepete kayıt işlemi sırasında hata oluştu"

        finally:
            return render_template("sonuc.html", msg = msg)
```

İd	Urun Adı	Urun Adet
2	Bardak	15

Satın Al

```
@app.route('/sepetlistele')
@login_required # bunu kapatmamalıyız dikkat kapatırsak insanın anlamayacağı
def sepetlistele():
    db_uri = 'sqlite:///basic_app.sqlite'
    engine = create_engine(db_uri)
    conn = engine.connect()
    select_st=select([table_sepet.c.id, table_sepet.c.urunadi,
        table_sepet.c.urunAdet]).where(table_sepet.c.user_id==current_user.id)
    rows = conn.execute(select_st)
    return render_template("sepetlistele.html", rows = rows)
```

Adeti değişecek Ürün id

Adeti

Değiştir

Silinecek Ürün id

Ürünü Sil

Sepeti Boşalt

```
@app.route('/Satis',methods = ['POST','GET'])
@login_required
def Satis():
    if request.method == 'POST':
        con = sqlite3.connect("basic_app.sqlite")
        idnu = current_user.id

        try:
            with con:
                sorgu = "insert into GecmisSepetler (urunadi,urunAdet,timestamp,user_id) "
                +"select sepetler.urunadi, sepetler.urunAdet, sepetler.timestamp, sepetler.user_id "
                |+"from sepetler where sepetler.user_id = ?"
                sorgu2 = 'delete from sepetler where user_id = ?'
                con.row_factory = sqlite3.Row
                cur = con.cursor()
                cur.execute(sorgu,(idnu))
                con.commit()
                cur.execute(sorgu2,(idnu))
                con.commit()
                msg = "Satın alma başarılı"
            except:
                msg = "Satın almada hata oluştu"

            finally:
                con.close()
                return render_template("sonuc.html",msg = msg)
```

[Anasayfaya dön](#)

```
@app.route('/UrunAdetDegistir', methods = ['POST','GET'])
@login_required
def UrunAdetDegistir():
    if request.method == 'POST':
        con = sqlite3.connect("basic_app.sqlite")
        try:
            urunid = request.form['urunid']
            urunAdet = request.form['urunAdet']

            with con:
                sorgu = '''update sepetler set urunAdet = ? where id = ? '''
                con.row_factory = sqlite3.Row
                cur = con.cursor()
                cur.execute(sorgu,(str(urunAdet),urunid))
                msg = "Güncelleme başarılı"

            except:
                con.rollback()
                msg = "Güncelleme başarısız"

            finally:
                con.close()
                return render_template("sonuc.html", msg = msg)
```

```
@app.route('/UrunSil', methods = ['POST','GET'])
@login_required
def UrunSil():
    if request.method == 'POST':
        con = sqlite3.connect("basic_app.sqlite")
        try:
            silinenid = request.form['silinenid']

            with con:
                sorgu = 'delete from sepetler where id = ? '
                con.row_factory = sqlite3.Row
                cur = con.cursor()
                cur.execute(sorgu,(silinenid))
                msg = "Silme başarılı"

        except:
            con.rollback()
            msg = "Silme başarısız"

        finally:
            con.close()
            return render_template("sonuc.html", msg = msg)
```

```
@app.route('/UrunleriSil', methods = ['POST','GET'])
@login_required
def UrunleriSil():
    if request.method == 'POST':
        con = sqlite3.connect("basic_app.sqlite")
        try:
            with con:
                sorgu = 'delete from sepetler'
                con.row_factory = sqlite3.Row
                cur = con.cursor()
                cur.execute(sorgu)
                msg = "Sepet başarılı ile boşaltıldı"

        except:
            con.rollback()
            msg = "Sepet boşaltma başarısız"

        finally:
            con.close()
            return render_template("sonuc.html", msg = msg)
```



## Geçmiş Alışveriş Listesi

İd	Urun Adı	Urun Adet	Urun Tarihi
1	Çanta	5	2018-12-11 16:09:05.605639
2	Uçlu Kalem	5	2018-12-11 16:09:05.605639

[Anasayfaya dön](#)

```
@app.route('/gecmissepet')
@login_required
def gecmissepet():
    db_uri = 'sqlite:///basic_app.sqlite'
    engine = create_engine(db_uri)
    conn = engine.connect()
    select_st=select([table_gecmissepet.c.id, table_gecmissepet.c.urunadi, table_gecmissepet.c.urunAdet,
        table_gecmissepet.c.timestamp]).where(table_gecmissepet.c.user_id==current_user.id)
    rows = conn.execute(select_st)
    return render_template("gecmissepetliste.html", rows =rows)
```

## Admin

### Admin Sayfası

Bu yalnızca adminlerin giriş yapabildiği bir sayfadır

Admin İşlemleri

İsim Girişi

Kayıt ol

Giriş Yap

Ana sayfa (Herkes erişebilir)

Üye sayfası (Giriş gerekli: member@example.com / Password1)

Admin sayfası (Rol gerekli: admin@example.com / Password1)

Çıkış

```
@app.route('/admin_page')
@roles_required('Admin') #Use of @roles_required decorator
def admin_page():
    return render_template_string("""
        {% extends "flask_user_layout.html" %}

        {% block content %}
            <h2>{{trans}}Admin Sayfası{{endtrans}}</h2>
            <h1>{{trans}}Bu yalnızca adminlerin giriş yapabildiği bir sayfadır{{endtrans}}</h1>
            <p><a href={{ url_for('admin') }}>{{trans}}Admin İşlemleri{{endtrans}}</a></p>
            <p><a href={{ url_for('isim_sayfasi') }}>{{trans}}İsim Girişi{{endtrans}}</a></p>
            <p><a href={{ url_for('user.register') }}>{{trans}}Kayıt ol{{endtrans}}</a></p>
            <p><a href={{ url_for('user.login') }}>{{trans}}Giriş Yap{{endtrans}}</a></p>
            <p><a href={{ url_for('home_page') }}>{{trans}}Ana sayfa{{endtrans}}</a> (Herkes erişebilir)</p>
            <p><a href={{ url_for('member_page') }}>{{trans}}Üye sayfası{{endtrans}}</a> (Giriş gerekli: member@examp
            <p><a href={{ url_for('admin_page') }}>{{trans}}Admin sayfası{{endtrans}}</a> (Rol gerekli: admin@example
            <p><a href={{ url_for('user.logout') }}>{{trans}}Çıkış{{endtrans}}</a></p>
        {% endblock %}
    """)
```

## Admin işlemleri

[Urun Ekle](#)

[Urun Güncelle](#)

[Urun Sil](#)

## Urun ekle

# Ürün Ekle

```
@app.route('/kayit_ekle', methods = ['POST', 'GET'])
def kayit_ekle():
    if request.method == 'POST':
        try:
            adi = request.form['urunadi']
            fiyati = request.form['urunfiyat']

            with sqlite3.connect("Data.db") as con:
                cur = con.cursor()
                cur.execute("INSERT INTO Urunler (urunadi,urunfiyat) VALUES (?,?)",(adi,fiyati) )
                con.commit()
                msg = "Kayit basari ile eklendi"
        except:
            con.rollback()
            msg = "Kayit islemi sirasinda hata olustu"

        finally:
            con.close()
            return render_template("sonuc.html",msg = msg)
```

## Ürün Güncelle

Bulma sonucu: Kayit bulundu

Seçim	Ürün Adı	Ürün fiyatı
<input type="checkbox"/>	Çanta	100

## Ürün bilgileri

[Ana sayfaya dön](#)

```
@app.route('/guncelle', methods = ['POST', 'GET'])
def guncelle():
    if request.method == 'POST':
        try:
            u_adi = request.form['ad']
            u_fiy = request.form['fiyat']
            secil = request.form["secilen"]

            with sqlite3.connect("Data.db") as con:
                sorgu = '''update Urunler set urunadi = ? , urunfiyat = ? where id = ? '''
                con.row_factory = sqlite3.Row
                cur = con.cursor()
                cur.execute(sorgu,(u_adi,u_fiy,secil[0]))
                msg = "Güncelleme başarılı"
        except:
            con.rollback()
            msg = "Güncelleme başarısız"

        finally:
            con.close()
            return render_template("sonuc.html", msg = msg)
```

Ürün Id	Ürün Adı	Ürün Fiyatı
1	Uçlu kalem	7
2	kağıt	2
3	silgi	10
4	su	1
6	asda	7
7	Bardak	20
8	Kitap	24
9	Çanta	100
10	Bilgisayar	2000

```
@app.route('/urunsilme')
def urunsilme():
    con = sqlite3.connect("Data.db")
    con.row_factory = sqlite3.Row
    cur = con.cursor()
    cur.execute("select * from Urunler")
    rows = cur.fetchall()
    return render_template("urunsilme.html", rows = rows)
```

Silinecek İd

[Anasayfaya dön](#)

### Tarihli arama

01.01.2018

[Anasayfaya dön](#)

### Bulma sonucu: Kayıtlar Bulundu

Urun Adı	Urun Adet	Tarih
Çanta	5	2018-12-11 16:09:05.605639
Uçlu Kalem	5	2018-12-11 16:09:05.605639

[Anasayfaya dön](#)

```
@app.route('/bultrh', methods = ['POST','GET'])
def bultrh():
    con = sqlite3.connect("basic_app.sqlite")
    con.row_factory = sqlite3.Row
    if request.method == 'POST':
        tarih = request.form['trh']
        try:
            with con:
                cur = con.cursor()
                cur.execute("select * from GecmisSepetler where timestamp > '"+tarih+"'")
                rows = cur.fetchall()
                msg = "Kayıtlar Bulundu"
        except:
            con.rollback()
            msg = "Kayıtlar bulunurken hata"

        finally:
            con.close()
            return render_template("tarihsonuc.html", msg = msg, rows=rows)
```

## Ürün Satış Bilgisi

Ürün Adı	Sayısı
Uçlu Kalem	1
Çanta	1

[Anasayfaya dön](#)

```
@app.route('/urunsatisbilgisi')
def urunsatisbilgisi():
    con = sqlite3.connect("basic_app.sqlite")
    con.row_factory = sqlite3.Row

    cur = con.cursor()
    cur.execute("select urunadi, Count(*) AS sayi FROM GecmisSepetler group by urunadi order by Count(*) DESC")

    rows = cur.fetchall()
    return render_template('urunsatisbilgisi.html', rows = rows)
```