

FURKAN ÇELİK H5170017

İLERİ JAVA FİNAL PROJE

Bu projede java spark (core spark-template-freemarker), apache derby, slf4j kütüphaneleri kullanılmıştır.

Proje içeriğinde basit bir alışveriş sitesi yapılmıştır. Alışveriş sitesi içerisinde 2 kullanıcı türü vardır. Bunlardan bir tanesi ürünleri satın alabilen, sipariş oluşturabilen normal kullanıcılar, diğer kullanıcı tipi ise ürün ve üye ekleyebilen yönetici kullanıcısıdır. Giriş yapan kullanıcının tipine göre 2 farklı anasayfa oluşturulmuştur. Kullanıcıların sayfaya erişim yetkileri sayfa yüklenirken kontrol edilmektedir. Eğer kullanıcının sayfaya erişim yetkisi yoksa kullanıcı hata sayfasına yönlendirilmektedir.

Uygulama "UygulamaMain.java" dosyası üzerinden başlatılmaktadır.

```
1 package spark.template.freemarker.furkancelik.finalproje;
2
3 import java.io.IOException;
4
5
6 public class UygulamaMain {
7
8     /**
9     * Final projesi uygulama maini
10    * @param args
11    */
12    public static void main(String[] args) {
13
14        Sunucu sunucu = new Sunucu();
15        sunucu.baslat();
16
17    }
18 }
```

Buradaki sunucu sınıfı içerisinde tüm spark linklerini ve ilk kurulum için gerekli fonksiyonları içermektedir.

```
@Override
public Object handle(Request request, Response response) {

    if(islem.equals("giris_goster")){
        return new ModelAndView(null, "/furkancelik/finalproje/giris_sayfasi.html");
    }

    if(islem.equals("giris_yap")){
        String kullanicAdi = request.queryParams("kullanicadi");
        String sifre = request.queryParams("sifre");

        Kullanici kullanici = sunucu.veritabani.KullaniciBul(kullanicAdi);

        // Kullanici bulundu sifre kontrolü yap
        if(kullanici != null){
            if(kullanici.getSifre().equals(sifre)){
                System.out.println("Kullanici giris yapti : " + kullanicAdi);
                sunucu.girisYapmisKullanici = kullanici;

                // Rol kontrolü yap
                if(kullanici.getKullaniciTipi().equals("ADMIN")){
                    response.redirect("/furkancelik/yonetici");
                }else{
                    response.redirect("/furkancelik/");
                }
            }else{
                sunucu.hata="Şifreniz kullanıcı ile uyuşmadı!";
                response.redirect("/furkancelik/hata");
            }
        }
    }
}
```

Problems @ Javadoc Declaration Console

Sunucu sınıfı içerisinde başlangıçta veritabanı işlemleri içerisindeki tablo oluştur fonksiyonu çalışmaktadır. Bu fonksiyon uygulama ilk çalıştığında ihtiyacımız olan tabloları veri tabanına eklemektedir. Eğer tablolar ekli ise eklememektedir.

Sayfa grupları için ayrı sınıflar tanımlanmıştır. Örn “GirisSayfasi.java” sınıfı 3 parametre almaktadır. Bunlardan ilki site üzerinden gelecek linktir. İkincisi ile oluşturulmuş sunucuyu temsil etmektedir. Sunucuyu ikinci defa oluşturunca hata veriyor. Bu sebeple oluşturulmuş sunucuyu “this” komutu ile bu sınıfla paylaşıyoruz. Ayrıca sunucu sınıfı içerisinde giriş yapan kullanıcı, kullanıcının siparişleri ve sunucudaki son hata verisi tutulmaktadır.

```
@Override
public Object handle(Request request, Response response) {

    if(islem.equals("giris_goster")){
        return new ModelAndView(null,"/furkancelik/finalproje/giris_sayfasi.html");
    }

    if(islem.equals("giris_yap")){
        String kullanciadi = request.queryParams("kullanciadi");

        String sifre = request.queryParams("sifre");

        Kullanici kullanci = sunucu.veritabani.KullaniciBul(kullanciadi);

        // Kullanici bulundu sifre kontrolü yap
        if(kullanci != null){
            if(kullanci.getSifre().equals(sifre)){
                System.out.println("Kullanici giris yapti : " + kullanciadi);
                sunucu.girisYapmisKullanici = kullanci;

                // Rol kontrolü yap
                if(kullanci.getKullaniciTipi().equals("ADMIN")){
                    response.redirect("/furkancelik/yonetici");
                }else{
                    response.redirect("/furkancelik/");
                }
            }else{
                sunucu.hata="Şifreniz kullanıcı ile uyuşmadı!";
                response.redirect("/furkancelik/hata");
            }
        }
    }
}
```

GirisSayfasi sınıfı içerisinde işlemlere göre sorgulama yapılmıştır. Kayıt olma giriş yapma gibi işlemlerin tamamı bu java dosyası üzerinden yapılmaktadır. Bu sınıfın bir diğer özelliği “IsimAbstract.java” sınıfıyla genişletilmesidir (Bu sınıf için yardım aldım). Bu sınıfın tek işlevi tüm girilen linklerin başına furkancelik eklemektedir. Sonrasında ise bir üst sınıf olan “FreeMarkerRoute” oluşturmasıdır.

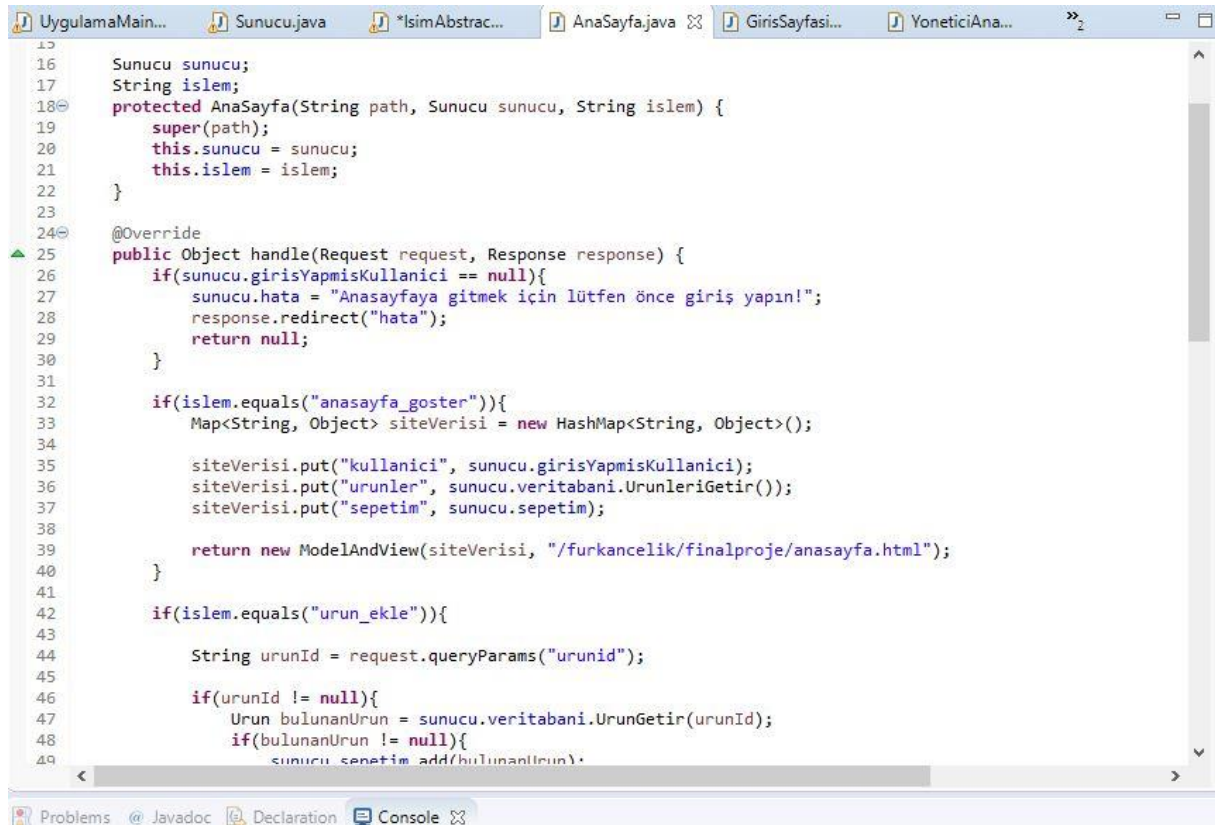
Kayıt olma işlemi için girilen verilerin eksiksiz olup olmadığı kontrol edilir ve veritabanına yazılır. Giriş yapma işlemi için girilen kullanıcı adı ve şifre veritabanında sorgulanır. Eğer eşleşiyorsa giriş yapan kullanıcı Sunucu.java dosyasına kaydedilir ve kullanıcı tipine göre bir anasayfaya yönlendirilir.

```

2
3 import spark.Request;
7
8 /**
9  * Isim abstract sınıfı tüm ürünlerin başına isim eklemek için kullandım
10  * Örneğin /(herhangi bir isim)/sayfa-adi
11  * Sparkın route classı ile extend edildi, sebebi her sınıf
12  * aslında sparkın bir sınıfı gibi davranacaktır.
13  */
14 public class IsimAbstract extends FreeMarkerRoute {
15
16     /**
17      * Bunu kullanan sınıfların bu fonksiyonu super ile çağırması gerekmektedir.
18      */
19     /**
20     * protected IsimAbstract(String path) {
21         super("furkancelik/" + path);
22     }
23
24     /**
25     * Bu method eğer sınıf içinden çağırılmamışsa tetiklenecektir
26     */
27     /**
28     * @Override
29     * public Object handle(Request request, Response response) {
30         return "Sayfa bulunamadı";
31     }
32 }
33

```

IsimAbstract sınıfı FreeMarkerRoute sınıfından türetilmektedir. Bu sınıf sayesinde websitesine gelen isteği yakalayabiliyoruz.

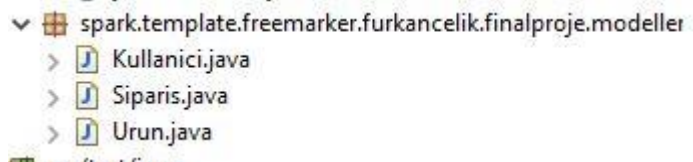


```

15 UygulamaMain... Sunucu.java *IsimAbstrac... AnaSayfa.java 2 GirişSayfası... YöneticiAna...
16 Sunucu sunucu;
17 String islem;
18 protected AnaSayfa(String path, Sunucu sunucu, String islem) {
19     super(path);
20     this.sunucu = sunucu;
21     this.islem = islem;
22 }
23
24 @Override
25 public Object handle(Request request, Response response) {
26     if(sunucu.girisYapmisKullanici == null){
27         sunucu.hata = "Anasayfaya gitmek için lütfen önce giriş yapın!";
28         response.redirect("hata");
29         return null;
30     }
31
32     if(islem.equals("anasayfa_goster")){
33         Map<String, Object> siteVerisi = new HashMap<String, Object>();
34
35         siteVerisi.put("kullanici", sunucu.girisYapmisKullanici);
36         siteVerisi.put("urunler", sunucu.veritabani.UrunleriGetir());
37         siteVerisi.put("sepetim", sunucu.sepetim);
38
39         return new ModelAndView(siteVerisi, "/furkancelik/finalproje/anasayfa.html");
40     }
41
42     if(islem.equals("urun_ekle")){
43
44         String urunId = request.queryParams("urunid");
45
46         if(urunId != null){
47             Urun bulunanUrun = sunucu.veritabani.UrunGetir(urunId);
48             if(bulunanUrun != null){
49                 sunucu.sepetim.add(bulunanUrun);

```

Anasayfa.java dosyasında normal kullanıcının yapabileceği işlemler bulunmaktadır. Burada anasayfa yüklenirken veritabanından ürünler ve Sunucu.java dosyasında tutulan sepetim listesi gösterilir. Bunun dışında sepete ürün ekleme, çıkarma ve sipariş oluşturma işlemleri bulunmaktadır. (HATA – Eğer sepette aynı üründen 4-5 tane varsa, ve birini çıkarmaya çalışırsam, aynı tipteki tüm ürünler çıkıyor, henüz nedenini çözemedim ve çözemedik bilginiz olsun).



Veritabanındaki her tablo için bir model oluşturdum (Bunu tavsiye sonrası yaptım). Basitçe buradaki modeller bir tablodaki bir satırlık kaydı tutuyor. Veri okurken yada veri yazarken bu modelleri fonksiyonlar içinden çağırıyorum.

```
2
3 public class Kullanici {
4
5     private String kullaniciAdi;
6     private String kullaniciTipi;
7     private Integer yas;
8     private String cinsiyet;
9     private String sifre;
10
11     public Kullanici(String kullaniciAdi, String kullaniciTipi, Integer yas, String cinsiyet, String :
12         this.kullaniciAdi = kullaniciAdi;
13         this.kullaniciTipi = kullaniciTipi;
14         this.yas = yas;
15         this.cinsiyet = cinsiyet;
16         this.sifre = sifre;
17     }
18
19     public String getKullaniciAdi() {
20         return kullaniciAdi;
21     }
22
23     public void setKullaniciAdi(String kullaniciAdi) {
24         this.kullaniciAdi = kullaniciAdi;
25     }
26
27     public String getKullaniciTipi() {
28         return kullaniciTipi;
29     }
30 }
```

```
2
3 public class Siparis {
4
5
6     public Siparis(int siparisId, String kullaniciAdi, String urunIdler, int toplamUcret) {
7         this.siparisId = siparisId;
8         this.kullaniciAdi = kullaniciAdi;
9         this.urunIdler = urunIdler;
10        this.toplamUcret = toplamUcret;
11    }
12
13    private int siparisId;
14
15    private String kullaniciAdi;
16
17    private String urunIdler;
18
19    private int toplamUcret;
20
21    public int getSiparisId() {
22        return siparisId;
23    }
24
25    public void setSiparisId(int siparisId) {
26        this.siparisId = siparisId;
27    }
28 }
```



```
YoneticiAna... Veritabanil... AnaSayfa.java Kullanici.java Siparis.java Urun.java ✕
1 package spark.template.freemarker.furkancelik.finalproje.modeller;
2
3 public class Urun {
4
5     private int urunId;
6     private String urunAdi;
7     private String urunKategori;
8     private int fiyat;
9
10    public Urun(int urunId, String urunAdi, String urunKategori, int fiyat) {
11        this.urunId = urunId;
12        this.urunAdi = urunAdi;
13        this.urunKategori = urunKategori;
14        this.fiyat = fiyat;
15    }
16
17    public int getUrunId() {
18        return urunId;
19    }
20
21    public void setUrunId(int urunId) {
22        this.urunId = urunId;
23    }
24
25
26
27
28
29
30
31
32
33
34    public VeritabaniIslemleri() {
35        try {
36            Class.forName("org.apache.derby.jdbc.EmbeddedDriver").newInstance();
37
38            Connection veritabaniBaglantisi = DriverManager.getConnection("jdbc:derby:"
39                + ".\\furkancelik10\\" // windows ortamı için
40                + ".\\furkancelik/"
41                + ";create=true;");
42
43            anaBaglanti = veritabaniBaglantisi.createStatement();
44
45        } catch (Exception e) {
46            System.out.println("Veritabani Baglantisi Hatasi : " + e.getMessage());
47        }
48    }
49
50    public boolean TablolariOlustur() {
51        try {
52            // Kullanıcılar tablosunu ekle
53            System.out.println("Kullanıcılar tablosu ekleniyor...");
54            anaBaglanti.execute("create table kullanicilar(" +
55                "kullanici_adi varchar(255)," +
56                "kullanici_tipi varchar(255)," +
57                "yas int," +
58                "cinsiyet varchar(255)," +
59                "sifre varchar(255))");
60            System.out.println("Kullanıcılar tablosu eklendi...");
61        }
62    }
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

VeritabaniIslemleri.java dosyası içerisinde veritabanıyla ilgili tüm işlemler bulunmaktadır. Kullanıcı ekleme/arama/silme ürün ekleme sipariş ekleme vb.

NOT: Yapamadığım yerlerde okulumuzun web sorunlusu Arda Bey'den yardım aldım