**05.04.2023**

İlgili günde kullanıcıların sayfalar arasında sunucu ile etkileşimli olarak gezinmesini sağlamak için gerekli yönlendirmeler ve Login (Giriş) ve Registration (Kayıt) isteklerinin sunucu tarafından kabul edilmesi için gerekli geliştirmeler yapılmıştır.

Kullanıcıların URL adresleri arasındaki gezinirken HTTP protokolüne ait olan GET metodu sunucuya gönderilmektedir. Gönderilen bu istekler sunucu tarafından kabul edilip yanıt olarak bir HTML sayfası döndürülmelidir. Bu isteklerin kabul edilip yanıt verilmesini sağlayan katman olan Controller katmanı oluşturulup bu katman içerisinde URL yönlendirmelerini ayarlayacak AppController isimli bir sınıf oluşturulmuştur. Bu sınıfın istekleri içeriye alabilmesi için @Controller anotasyonu ile işaretlenmiştir.

Kullanıcı <http://localhost:8080> gibi alt adresi olmayan bir URL adresi ile sunucudan istekte bulunduğu zaman giriş sayfasına yönlendirilmesi için aşağıdaki fonksiyon AppController sınıfı içerisinde tanımlanmıştır. “” ifadesi alt adres olmadığını belirtmektedir. ‘return “login”’ ifadesi kullanıcıya login.html dosyasının return edilmesi gerektiğini belirtir. Login ile ilgili geriye kalan bütün işlemler (authentication) Spring Security tarafından işlenmektedir.

@GetMapping("")

public String viewLoginPage() {

return "login";

}

Kullanıcının “Kayıt ol” sayfasına erişebilmesi için AppController sınıfında aşağıdaki tanımlama yapılmıştır. Kullanıcı <http://localhost/register> adresi ile etkileşime girdiği zaman bu fonksiyon çalışmaktadır. Kullanıcı girmiş olduğu username ve parola bilgisinin kaydetme işlemi sırasında alınabilmesi için yeni bir User nesnesi sayfaya gömülerek gönderilmektedir.

@GetMapping("/register")

public String showRegistrationForm(Model model) {

model.addAttribute("user", new User());

return "signup";

}

Kullanıcının “Kayıt ol” sayfasında bilgilerini doldurup işlemi onayladığı zaman bilgilerin alınıp kaydetme işleminin gerçekleşebilmesi için aşağıdaki fonksiyon tanımlanmıştır. POST yöntemi ile gelen ve /process\_register adresine yapılan istekler bu sınıf tarafından kabul edilir. İstek geldiği durumda kullanıcının girmiş olduğu username ve password bilgisi parametre olarak gelen user nesnesinde otomatik olarak set edilmiş durumdadır. Parola bilgisi BCrypt algoritması ile şifrelenip, user nesnesine şifreli hali set edilip ve Repository katmanı aracılığıyla veritabanına kaydedilmektedir. Kayıt işlemi gerçekleştikten sonra kullanıcıya kaydın başarıyla gerçekleştirildiğini söyleyen bir sayfa gösterilmektedir.

@PostMapping("/process\_register")

public String processRegister(User user) {

BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();

String encodedPassword = passwordEncoder.encode(user.getPassword());

user.setPassword(encodedPassword);

userRepo.save(user);

return "register\_success";

}

=====

Kullanıcı login işlemini başarıyla tamamladıktan sonra kullanıcının dosya yükleme sayfasına yönlendirilmesi için aşağıdaki fonksiyon tanımlanmıştır. “/home” adresine yönlendirme Spring Security’nin konfigürasyon ayarlarında tanımlanan “defaultSuccessUrl” tanımı sayesinde yapılmaktadır.

@GetMapping("/home")

public String home (Model model) {

return “home”;

}