Uygulama

Hedef: MVC tabanlı Restful mimariye sahip, Bootstrap/Jquery entegre edilmiş bir uygulama gerçekleştirmek. Uygulamada hedef, kitap adı, özeti ve resmi girilebilen bir kitap uygulaması ara yüzü ve yazar girilebilen bir yazar ara yüzü oluşturmak. Uygulama ya devise gem kullanarak kullanıcı yönetimi ekleyeceğiz ve bir sonraki hafta Yazar-User arasında 1-1, Yazar-Kitap arasında ve User-Kitap arasında 1-Many ilişki kuracağız.

Ana adımlar

1) Uygulama template'i oluşturma
2) Scaffold kullanarak kitap kaynağı için MVC dosyalarını üretmek
3) Root sayfası verme
4) Alias verme
5) Bootstrap/jquery entegre etme
6) Paperclip gem ile kitap uygulamasına resim eklenmesi
7) Uygulamaya ara yüz oluşturma

Bölüm sonunda: Tamamen farklı bir uygulama için (örneğin stok takibi için ürün adı, miktarı, fiyatı, resmi girilen bir uygulamayı baştan sona yapabilmeniz gerekmektedir)

Adımlar.

1) Uygulama şablonu oluşturma

   rails new uygulama
   cd uygulama

2) Scaffold ile çalışan bir /kitaps restful mvc uygulaması üretmek (Bu scaffold ile aynı zamanda Kitap isimli bir tablo oluştuğunu hatırlayalım)

   rails g scaffold kitap isim ozet:text

   rake db:migrate

   rails s

3) Root (açılış sayfası verme)
   #config/routes.rb #ROR ortamında tüm pathler bu dosyadan ayarlanır
   #aşağıdaki satırı üstteki dosyaya ekleyerek kitaps controllerinin (app/controller/kitaps_controller) index #methodunu açılış sayfası yap demiş oluyoruz

   root "kitaps#index"

4) Alias verme (Linkleri takma isimle yönlendirmek)

   #config/routes.rb içinde ki resources :kitaps satırı aşağıdaki gibi düzenlenecek

   resources :kitaps, :path=>"kitaplar"

Diğer adımlara başlamadan önce herhangi bir gem ne işe yarar, nasıl kurulur/konfigure edilir, tüm gemler için bir arama motoru ve yönetim sistemi olan rubygems.org adresinden bakabilirsiniz.

5) Uygulamaya bootstrap ve jquery desteği eklenmesi

#Gemfile a aşağıdaki kütüphaneler eklenecek

gem 'bootstrap'

gem 'jquery-rails'

#Komut satırında, uygulama klasörü içinde aşağıdaki komut çalıştırılacak (Gemfile eklenen yeni kütüphanelerin indirilme ve kurma işlemi gerçekleştirilir)

bundle

#app/assets/stylesheets/application.css dosyası

#application.scss olarak değiştirilir

#ardından application.scss dosyasında en alta aşağıdaki satır eklenir

@import "bootstrap";

#app/assets/javascripts/application.js

#dosyasına aşağıdaki iki satır eklenir //=require . satırından önce olacak şekilde eklenecek

//= require jquery

//= require bootstrap

Not 1: Bootstrap/jquery kurulumundan sonra Puma sunucusu (rails s) açık ise Ctrl+C ile kapatarak yeniden başlatmanız gerekmektedir.

Not 2: Bootstrap/Jquery çalıştığını denetlemek için herhangi bir bootstrap öğesi (grid, jumbotron vb) sayfaya koyarak deneyiniz.

6) Paperclip gem ile kitap modeline resim eklemek

#Gemfile a aşağıdaki satır eklenecek ardından terminal ortamında uygulama klasörü içinde terminal ortamında bundle yapılacak

gem "paperclip"

bundle

Aşağıdaki satır (terminal ortamında) paperclip kullanarak Kitap modeline resim isminde bir alan ekler

rails g paperclip kitap resim

rake db:migrate #veritabanında ki değişikliklerin gerçekleştirilmesi

#app/controllers/kitaps_controller permit listesi güncellenecek

#kitap uygulaması en başta scaffold ile oluşturulmuştu ve permit (izin) listesinde sadece isim ve özet alanları vardı

params.require(:kitap).permit(:isim, :ozet, :resim)

#app/views/kitaps/_form.erb.html dosyasına dosya yüklenmesi için gerekli field eklenecek

** **

<%= form.label :resim %>

<%= form.file_field :resim %>

** **

#app/models/kitap.rb dosyasına resim dosyalarını eklemesi için

has_attached_file :resim, styles: { medium: "300x300>", thumb: "100x100>" }, default_url: "/images/:style/missing.png"

validates_attachment_content_type :resim, content_type: /\Aimage\/.*\z/

#resimleri göstermek için görünüm dosyaları editlenecek

#app/views/kitaps/index.html.erb

<%= image_tag kitap.resim.url(:thumb) if kitap.resim? %>

#app/views/kitaps/show.html.erb

<%= image_tag @kitap.resim.url(:medium) if @kitap.resim? %>

7) Arayüz için bootsnipp yada w3schools gibi kaynaklardan ücretsiz şablonlar seçebilirsiniz, şablon seçiminde kullanılan bootstrap ve jquery versiyonlarının sizin uygulamanız ile uyumlu çalışması ve farklı tarayıcılarda denenmesi gerekmektedir.

Biz 5.adımda bootstrap kurarken Gemfile a gem 'bootstrap' satırı ile ekleme yaptık, burada versiyon belirtmediğimiz için default olarak bootstrap 4 versiyonu yüklenir. Bu yüzden template bootstrap 4 yada uyumlu bir versiyon olması gerekmektedir.

Kullanılan örnek template linki https://bootsnipp.com/snippets/1eyWZ

Template nasıl uygulamaya eklenir.

Bootsnipp sitesindeki şablonlar ***Preview, Html, CSS, JS*** kodlarını içermektedir.

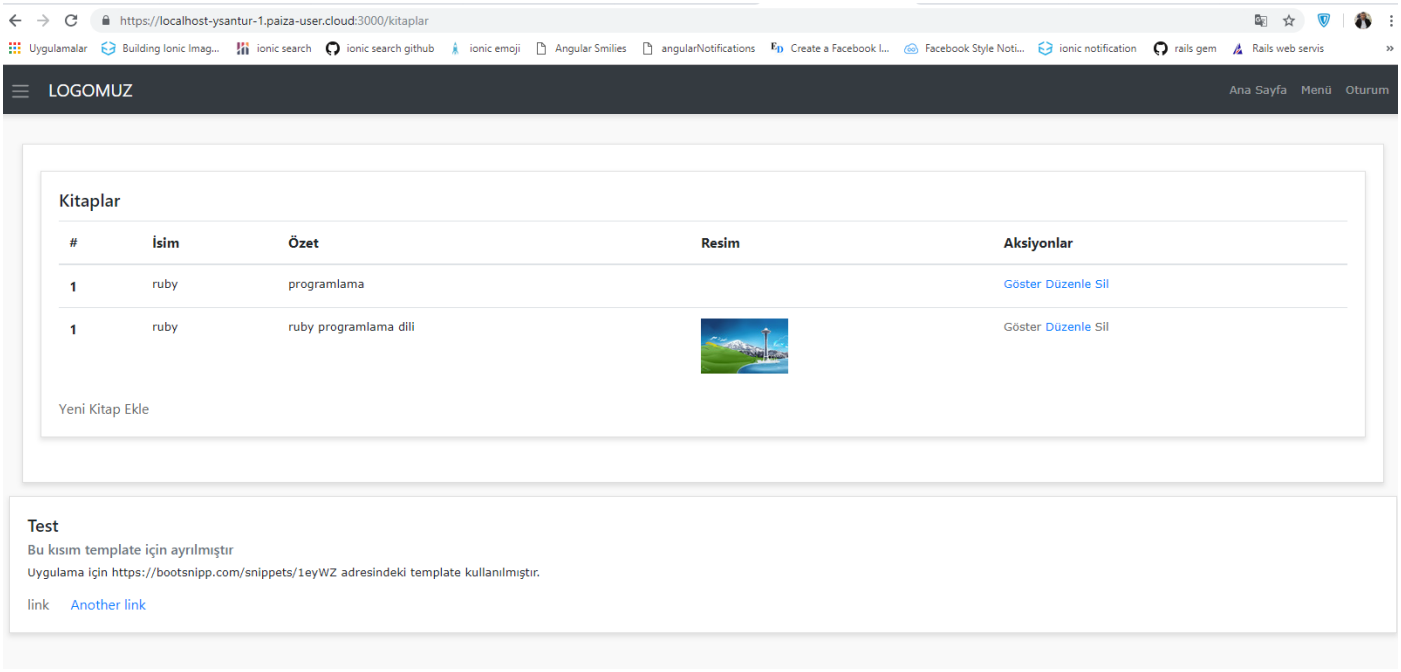
1) Html kodları uygulama layout dosyasına ***app/views/layouts/application.html.erb*** kopyalanır. Yalnız üstteki meta taglar kopyalanmasına gerek yoktur. Yani yukarıdaki örnek için 6.satırından itibaren bizim application.html.erb dosyasındaki body içine kopyalıyoruz.

   Not: application.html.erb dosyasındaki **<%=yield%>** satırı silinmemelidir.

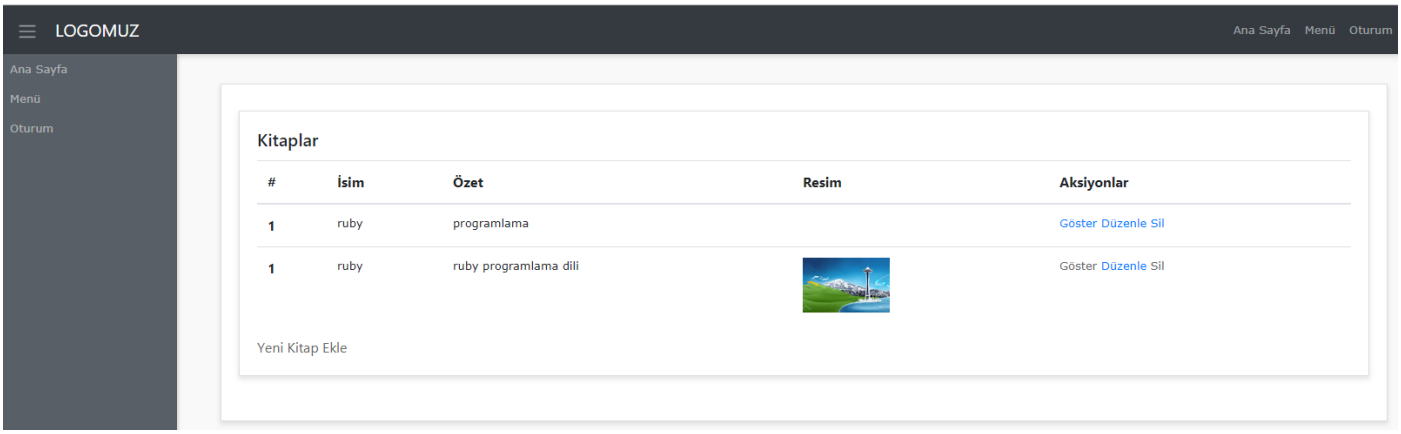
2) CSS kodları ***app/assets/stylesheets/application.scss*** dosyasına eklenir
3) JS kodları ***app/assets/javascripts/application.js*** dosyasına eklenir
4) Kullanılan şablon da ki html içerikler düzenlemek (gereksiz divleri kaldırmak, İngilizce içerikleri temizlemek/değiştirmek vb…) ve uygulamaya uygun hale getirmek

Üstteki 7 adımı tamamlayarak, aşağıdaki gibi ekran görüntülerinde olduğu gibi grid sistemide entegre ederek uygulamayı çalışır hale getirin.

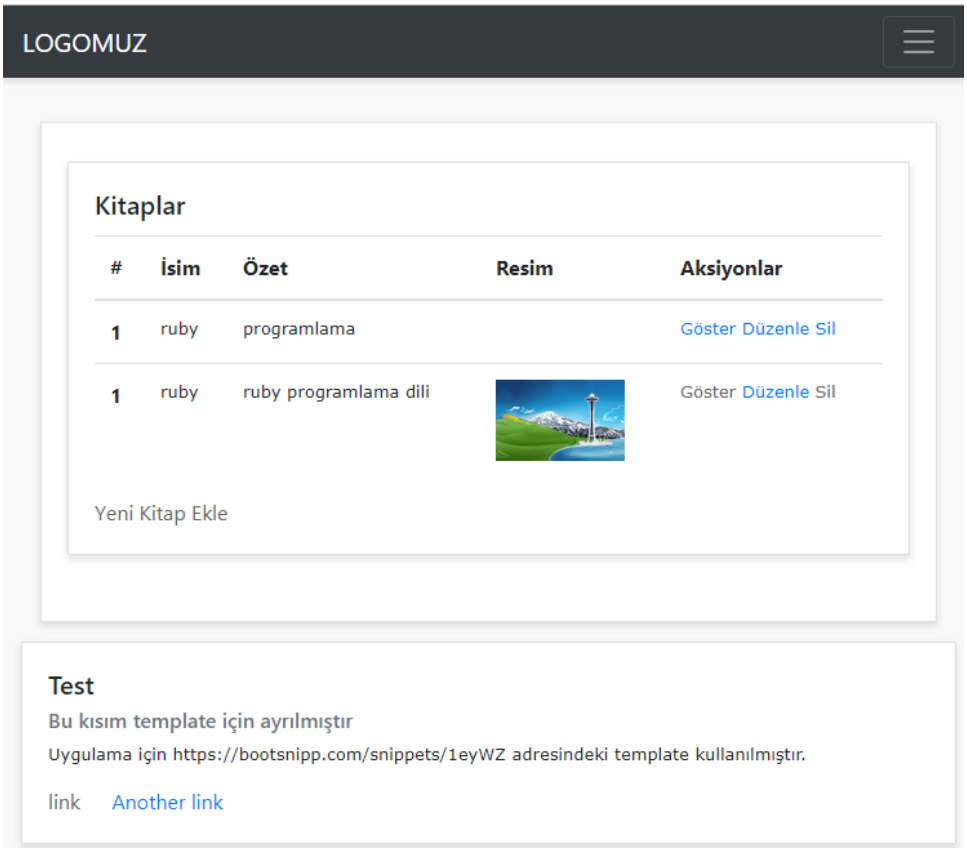
Görsel 1 => Uygulama ana sayfa görünümü (Grid sistem yerleştirilmiş template uygulamaya uyarlanmış ve Türkçeleştirilmiş)



Görsel 2 => Side menü



Görsel 3 => Responsive



8) Scaffold ile çalışan bir /yazars restful mvc uygulaması üretmek (Bu scaffold ile aynı zamanda Kitap isimli bir tablo oluştuğunu hatırlayalım)

   rails g scaffold yazar isim

   rake db:migrate

   rails s

9) Yazars için Alias verme (Linkleri takma isimle yönlendirmek)

   #config/routes.rb içinde ki resources :yazars satırı aşağıdaki gibi düzenlenecek

   resources :yazars, :path=>"yazarlar"

Yazar uygulaması (/yazarlar) içinde ara yüzleri düzenleyin

DEVISE GEM İLE KULLANICI YÖNETİMİNİN UYGULAMAYA ENTEGRE EDİLMESİ

10) #Gemfile a aşağıdaki satır eklenecek ardından terminal ortamında uygulama klasörü içinde terminal ortamında bundle yapılacak

gem "devise "

bundle

11) Terminal ortamında, uygulama klasörü içinde sırası ile aşağıdaki komutlar çalıştırılarak devise geminin kurulumu yapılmış olacak. Bu işlem sonucunda email, password alanlarından oluşan **User** isimli bir modele sahip olmuş olacağız ***rails console*** ortamında bu tabloyu inceleyebilirsiniz.

rails g devise:install

rails g devise user

rails g devise:views

rake db:migrate

12) Kullanıcı yönetimi için navbarı bir **partial** olarak ekleyeceğiz. Sayfa html kodlarının tamamının ***application.html.erb*** içinde yazılmasından ziyade bir kısmı başka bir isimle kaydedilmesine partial adı verilir, partial isimleri alt çizgi (_) ile başlamak zorundadır!

12.1) Aşağıdaki kodları **app/views/shared/_navbar.html.erb** isimli bir dosya oluşturarak kaydedin. (Dosya adında alt çizgi kullanıldığına dikkat edin)

12.2) Ardından bu dosyadaki kodların uygulamaya dahil edilmesi için **app/views/layouts/application.html.erb** dosyası içinde grid sistemde uygun yere (7.maddede yapılan)

<%= render 'shared/navbar' %> satırı ile ekleyin. (link verirken alt çizgi kullanılMAdığına dikkat edin)

```
<p class="navbar-text pull-right"> <% if user_signed_in? %>

Logged in as <strong><%= current_user.email %></strong>.
<%= link_to 'Edit profile', edit_user_registration_path, :class => 'navbar-link' %> |
<%= link_to "Logout", destroy_user_session_path, method: :delete, :class => 'navbar-link' %>

<% else %>
<%= link_to "Sign up", new_user_registration_path, :class => 'navbar-link' %> | <%= link_to
"Login", new_user_session_path, :class => 'navbar-link' %>

<% end %> </p>
```

Notlar

current_user Sistemideki aktif kullanıcıyı simgeler

if user_signed_in? Sistemde giriş yapmış bir kullanıcı varmı True/False döner

Sign_up/login/Edit/Logout linklerine dikkat ediniz. (Ezberlemek zorunda değilsiniz terminal ortamında **rake routes** komutu size tüm linkleri verir)

13) Son olarak **app/controllers/kitaps_controller.rb** dosyası içinde aşağıdaki satırı yazarak controllere aksiyondan önce kullanıcı giriş şartı ara demiş oluyoruz.

before_action :authenticate_user!, only: [:new, :edit, :destroy] #yeni/güncelleme/silme için üyelik şartı ara

veya

before_action :authenticate_user!, except: [:index, :show] #index ve Show hariç üyelik şartı ara

Notlar

1) Devise gem i ile ilgili tüm görünümler (kayıt/login/doğrulama vb…) aşağıdaki dizindedir. Grid sistemi burada ki .html.erb dosyalarına da uygulamanız gerekmektedir.

app/views/devise

2) Devise ile ilgili ayarlar app/config/initializers/devise.rb dosyası içinde bulunur. Örneğin aşağıdaki satır şifrelerin en az, en çok kaç karakter olacağını belirler, 2.satır ise şifre politikası için regex paterni tanımlar

config.password_length = 6..128

config.email_regexp= /\A[^@\s]+@[^@\s]+\z/

ÖDEV

Bir sonraki hafta Yazar-User arasında 1-1, Yazar-Kitap arasında ve User-Kitap arasında 1-Many ilişki kuracağız. Bu nedenle **mutlaka** bu uygulamanın (kullanıcı yönetimi, grid sistemi entegre edilmiş, /kitaplar, /yazarlar şeklinde çalışır şekilde, 19.04.2019 pazartesi günkü derse gelmeniz gerekmektedir. Ayrıca aşağıda ki gözden geçirme sorularının cevaplarını çözmüş olarak gelin.

Gözden geçirme soruları

1) Uygulama çalışan tüm linkleri nasıl görürsünüz.
2) Kullanıcının login/signup olmasını sağlayan görünüm dosyalarının tam yolu nedir.
3) **=link_to** erb kodu ne işe yarar. Bu kodu kullanarak kullanıcıya logout/login/signup yaptırma linklerini yazınız.
4) Kullanıcının login/signup yollarının url linki nasıl yazılır. (Tarayıcıda görünen adres)
5) **current_user** aktif kullanıcı bilgisini nasıl çeker, cevaplayın.
6) Uygulama üye olan ilk kullanıcıyı, son kullanıcıyı nasıl çekeriz.
7) Web uygulamasına yönelik bir admin (yönetici) paneli nasıl hazırlanır, avantaj ve dezavantajları nedir.
8) Web uygulamasında çoklu dil desteği nasıl verirsiniz, mantığını kurgulayın.
9) Bu uygulama için bir arama sayfası oluşturmak istiyoruz (kitap ve/veya yazar ismine göre arama yapacağız), mantığını kurgulayın.