UNICORN COLLEGE

WEB Domácí úkol - 2. Kontrolní bod - Frontend (kombinovaná forma)

Kontext

Semestrální práce z předmětu WEB má 3 části (dva kontrolní body a finální odevzdání). Ke každé části se váže jeden artefakt (jedno zadání).

1. V první části si ověříte svoji schopnost práce s HTML a CSS (někteří i JS). Vaším cílem bude vytvořit landing page pro váš produkt. Odevzdáním této části si rovněž zvolíte téma své semestrální práce.
2. Ve druhé části (to jest tato) již budete pracovat v Ruby on Rails a vytvoříte tzv. modely a frontend vašeho produktu (pokud budete dělat například blog, tak v této části vytvoříte tu jeho část, kterou vidí vaši čtenáři). Poznámka: frontend má jiný (leč možno související) vzhled, tedy design než-li landing page!
3. Ve třetí a poslední části pak vytvoříte tzv. backend pro váš produkt, tedy administrační rozhraní pomocí kterého můžete svůj produkt spravovat (tedy pro náš příklad blogu můžete v administraci přidávat a upravovat např. články, kategorie a tagy vašich článků na blogu).

Každá webová aplikace potřebuje frontend - tedy tu část, kterou vidí její uživatelé. Proto jej bude mít i vaše semestrální práce. V tomto kontrolním bodu již budeme používat framework Ruby on Rails.

Abychom mohli frontend pro naši aplikaci vytvořit, potřebujeme mít i co v něm zobrazovat. Proto v tomto kontrolním bodě vytvoříme kromě frontendu i modely naší aplikace.

Zadání

Úkolem tohoto kontrolního bodu je vytvoření frontendu a modelů pro vaši semestrální práci.

Zpracování tohoto kontrolního bodu má tedy dvě čáti, které je nutné dělat v daném pořadí:

1. Vytvoření modelů aplikace
2. Vytvoření frontendu aplikace

Frontend aplikace již má zobrazovat reálná data (načtená z modelů). Proto je NUTNÉ nejprve vytvořit modely, poté až frontend.

Vytvoření modelů

Bez ohledu na rozsah vaší práce je třeba vytvořit aplikaci, která bude pracovat alespoň se třemi modely (mezi než nepočítáme model User). Mezi modely musí být alespoň jedna 1:N a alespoň jedna M:N asociace.

Jaké modely budete spravovat záleží na tématu vaší semestrální práce.

V tomto kontrolním bodě je třeba splnit (s ohledem na modely) následující:

- Vytvořit alespoň 3 modely (jiné než User)
- Mezi dvěma modely musí být vazba 1:N (has_many + belongs_to)
- Mezi dvěma modely musí být vazba M:N (has_many_throught, nikoli has_and_belongs_to_many!!)
- Aplikace musí mít seeds.rb, které ji naplní dostatečnými daty pro otestování frontendu

- Seedy vaší aplikace MUSÍ být idempotentní (tedy musí být možné provést rake db:seed opakovaně BEZ duplikace dat) - používejte find_or_create a nebo kontrolu toho zda data v DB už máte
- V seedech používejte vždy vykřičníkové varianty příkazů (např. create! místo create) - díky tomu se dozvíte, že se něco pokazilo

V tomto kontrolním bodu není třeba řešit:

- Cizí klíče
- Deletion strategy (co se má stát se souvisejícími záznamy, když je smazán na ně navázaný záznam - např. zda se mají produkty v kategorii eshopu smazat při smazání kategorie nebo ne)

U kontrolních bodů prosím používejte **POUZE sqlite databázi**! Jiné databáze s důvodu snadnosti kontroly nebudou akceptovány.

Pro vytvoření modelů používejte rails generate model.

Pro vytvoření migrací používejte rails generate migration. Migrace pouzivejte správně! **Nikdy je po aplikace NEMĚŇTE**, raději vždy vytvořte novou migraci.

Volitelně můžete k modelům jako BONUS vytvořit unit testy:

- Buďto pomocí nástroje RSpec: http://rspec.info/
- Nebo pomocí Cucumber: https://cucumber.io/docs/reference/ruby a https://cucumber.io/docs/reference/rails a https://github.com/cucumber/cucumber/wiki/A-Table-Of-Content

Vytvoření frontendu

V tomto kontrolním bodě je třeba splnit (s ohledem na frontend) následující:

- Vaše aplikace musí mít vyřešený vzhled - tedy CSS (dle varianty vaší semestrální práce)
- Frontend musí zobrazovat všechny modely, které vaše aplikace používá logickým způsobem (příklad, pokud je vaše aplikace blog, měl být na frontendu vidět seznam kategorií, kde když kategorii rozkliknu tak uvidím příspěvky do ní náležející, dále bych měl být schopen zobrazit si všechny příspěvky ze všech kategorií. Všechny příspěvky by v tomto příkladě měly být seřazeny dle data. Po rozkliknutí některého příspěvku bych se měl dostat na jeho detail, kde uvidím jeho název a text, stejně jako kategorii i seznam tagů. Když rozkliknu tak, měl bych vidět seznam příspěvků co mají tento tag).
- Frontend musí mít minimálně 3 stránky (nějaký seznam, nějaký detail a nějaký seznam položek, který jsem rozklikl z detailu a zobrazuje položky, které s položkou odkud jsem jej rozklikl souvisí.

V tomto kontrolním bodu není třeba řešit:

- Přidávání, úpravy a správu položek (to bude mít na starosti administrace, kterou budeme dělat v dalším kontrolním bodě) - pokud chcete i svým frontendovým uživatelům umožnit přidávání nebo správu obsahu, je to možné, ale v administraci to budete muset mít rovněž.
- Přihlašování a registraci (ta bude součástí administrace - pokud chcete umožňovat přihlašování i svým uživatelům, můžete si zatím připravit a nastylovat formulář).
- Vyhledávání

Tipy:

- aplikaci si určitě vygenerujte pomocí rails new
- vyhněte se použití rout match, místo toho používejte collection a member v resource, případně zanořené resource (viz. guides stránka "Rails routing from the outside in").
- migrace nikdy nepište sami, VŽDY je generujte - už kvůli názvu souboru (ano, může se vám stát že to nevygeneruje obsah té migrace, to nevadí, pak ho musíte dopsat - pokud to jde, chcete použít jen metodu change místo up & down)
- migrace nikdy neupravujte po spuštění rake db:migrate. Raději udělejte migraci novou. Pokud si nejste jisti zda ji upravit můžete nebo ne, tak je neupravujte nikdy.

Poznámky:

- Pro tento kontrolní bod NELZE používat gemy jako jsou rails_admin, active_admin, atd., ani nástroje jako jsou AlchemyCMS, RefineryCMS, Publify, apod. - použití těchto gemů, nebo nástrojů = 0 bodů, nebo nutnost přepracovat úkol!
- Je NUTNÉ používat Rails verze 4 a vyšší. Je možné použít Rails 5, ale materiály počítají z Rails 4.
- Je NUTNÉ dodržovat téma vaší semestrální práce!

V minulém kontrolním bodě jste se rohodli pro jednu ze dvou variant své semestrální práce. Níže jsou další podmínky zadání dle této varianty. Až po tento řádek se zadání týkalo obou variant.

Omezení pro variantu ze zelené louky

- Vaše aplikace musí být validní HTML, dle validátoru z minulého kontrolního bodu

Omezení pro variantu s frameworky

- Vaše aplikace musí být validní HTML5 včetně správného použití HTML5 formulářových prvků a sémantických tagů
- Vaše aplikace musí být responsivní

Bonusové úkoly:

Pokud máte pocit, že je tento kontrolní bod příliš triviální, můžete dobrovolně zpracovat následující:

- zkuste do aplikace přidat autentizaci pomocí gemu devise
- zkuste do aplikace přidat autorizaci pomocí gemu cancancan (že některý uživatel uvidí jen něco)

Postup

Při realizaci nejprve vytvořte modely, poté seedy (abyste získaly data) a poté výpisy na frontendu.

Nabízí se otázka "jak přidám/vytvořím záznam v modelu" když nemám formulář? Odpověď je pomocí rails console. Toto napíšete a do příkazové řádky můžete psát příkazy ActiveRecord Query Interface - tedy např. Article.create!(name: "Foo") ...

Řešení

Odeslat řešení domácího úkolu

Vaše řešení MUSÍ jít spustit "z ničeho" následujícím postupem (předpokládejte, že cvičícímu už funguje ruby a bundler). Tento postup bude kontrolující cvičící aplikovat a pokud mu neprojde, tak vaše práce bude ohodnocena 0 body!

Postup kontroly:

1. Stáhnutí .zip a jeho rozbalení, výstupem tohoto rozbalení by měl být 1 adresář
2. Přepnutí do tohoto adresáře
3. bundle install
4. rake db:migrate
5. rake db:seed
6. rails server
7. Pak se cvičící podívá na 0.0.0.0:3000, kde by mělo něco být (windowsáři se dívají na 127.0.0.1:3000)

Ve vlastním zájmu si, se svým zipem připraveným k odevzdání, tento postup vyzkoušejte provést! A až teprve potom co vám to PROJDE své řešení odevzdávejte!

I když úlohu odevzdáte, je možné za ni získat 0 bodů, toto nastane když:

- A) Obecné podmínky - 0 bodů máte když:

- Odevzdáte méně než 50% řešení
- Zjistíme, že jste úkol opsali (0 bodů dostane ten kdo řešení opsal i ten kdo řešení poskytl) - ANO, děláme diff - http://cs.wikipedia.org/wiki/Diff
- Odevzdáte úkol po DEADLINE, 1 minuta po DEADLINE je také po DEADLINE
- Odevzdáte úkol ve špatném formátu, se špatnou přílohou, atd. sice včas, ale logicky nebude akceptován. A pak nestihnete odevzdat opravenou verzi úkolu všas (do deadline). Toto NENÍ chyba cvičícího, že vám řešení "nestihl" opravit, ale VAŠE neboť jste nezvládli řešení odevzdat ve správném formátu napoprvé, nebo včas navzdory tomu, že je zde detailně popsán.

- B) Specifické podmínky tomuto předmětu či úkolu - 0 bodů máte když:
  - Zjistíme, že jste úkol opsali stylem copy paste z jiné webové stránky (zjistit toto je ještě snažší než diff)
  - Použijete tag <marquee>
  - Použijete font Comic Sans MS
  - Odevzdáte aplikaci, u které selže bundle install, rake db:migrate a nebo se aplikace nespustí (selže rails server).

Odevzdání JE BEZPODMÍNEČNĚ NUTNÉ provést přes systém UU (Unicorn Universe), a to NÁSLEDOVNĚ:

Během realizace tohoto úkolu by měl vzniknout adresář s Ruby on Rails aplikací, adresář pojmenujete:

ucl_web_02_novak_jan

ten zazipujete a vznikne vám jeden .zip (zazipujete CELÝ ADRESÁŘ, ne jen jeno obsah!). Tento zip pojmenujete "ucl_web_02_novak_jan.zip". Pokud se soubor jmenuje jinak, přejmenujete ho.

Místo novak_jan je pochopitelně VAŠE jméno BEZ diakritiky. ANO, ty divné čáry v tom názvu NEJSOU mezery, ale znaky podtržítka - "_"

Tento SPRÁVNĚ POJMENOVANÝ .zip odevzdáte do UU jako přílohu řešení. A POTÉ napíšete do POPISU řešení na UU:

Odevzdávám ucl_web_02_novak_jan.zip

Místo novak_jan je pochopitelně VAŠE jméno BEZ diakritiky.

Tedy správné odevzdání = správně pojmenovaná příloha + správně napsaný text popisu. Pokud chcete cvičítímu něco vzkázat, připíšete to do popisu úkolu, za "Odevzdávám ucl_web_02_novak_jan.zip", přičemž mezi "Odevzdávám ucl_web_02_novak_jan.zip" a vaším textem bude jeden prázdný řádek.

Pokud to uděláte jinak, VAŠE ŘEŠENÍ NEBUDE AKCEPTOVÁNO a budete to muset udělat znovu.

Pokud máte k úkolu, jeho zpracování, či odevzdání nějaké dotazy, kontaktujte VČAS, ne naposlední chvíli, svého cvičícího komunikačním kanálem, který preferuje a který vám pro tyto účely definoval.

Zdroje

Když nevíte, jak to implementovat, podívejte se na:

- Přednášky, přesně to, co se po vás chce, přednášející ukazoval na přednášce
- Web http://guides.rubyonrails.org - tam, JE vše co se po vás chce napsáno. Pokud ste to nepochopili z přednášek, pak si přečtěte tento web. Když si ho přečtete, tak to z něj nelze nepochopit. Pokud si ho přečtete a přesto nepochopíte, tak to znamená, že jste ho nepřečetli.