

Obecne informacie:

- Ak su pre riesitela zadania [1,2] jednoduche a vie podobny scope funkcionality demonstrovat na niecom vlastnom, staci odovzdat zadania: [3,4,5]
- Minimalne je ale potrebne odovzdat zadania [1,2]
- Zadania sa mozu riesit na blade sablonach alebo formou REST API s vue.js (alebo ine)
- Pri zadani c.1 sa moze riesitel rozhodnut ci bude riesit zadanie ajaxom (B) alebo nie (A)
- Flash spravy, ich existenciu a obsah (hlasky pri uzivatelskych akciach) ponechavame cisto na riesitela.

!! Kazde jednotlivé zadanie (jeho funkčný výsledok) musí byť v gite otagované patricnou verzioú. Cize pre zadanie c.1 bude výsledok otagovaný verzioú 0.1, pre zadanie c.2 bude tag 0.2 atď.

Laravel zadanie c.1

Vytvorit jednoduchú aplikáciu s 2 controllermi a modelmi, ďalej už len popis požadovanej funkcionality.

Je potrebne administracne rozhranie pre spravu knih. V menu bude dostupne pod zalozkou s nazvom "Books" rozhranie pre listing a administráciu danyh položiek "knih". Ďalej bude dostupne tlačitko <ADD>, na ktore keď užívateľ klikne stane sa:

A) Stranka sa presmeruje na formular pre vytvorenie noveho zaznamu "knihy"

B) Otvori sa modal okno s formularom pre vytvorenie a po ulozeni sa stranka AJAXOM refreshne

Pri kliknutí na názov alebo kliknutí na akciu UPDATE, sa užívateľovi objavi taktiez formular s možnosťou editácie buď podľa bodu A) alebo B).

Ďalej bude dostupna možnosť mazania zaznamu knihy pri kliknutí na akciu DELETE. Taktiez je potrebne implementovat strankovanie, kde na jednu stránku sa bude vypisovat [N] zaznamov. Hodnota [N] bude konfigurovatelna podľa .env premennej s nazvom "BOOKS_PER_PAGE" (integer)

Priklad:

<ADD>

ID	NAME	DESCRIPTION	PAGES	YEAR	ACTION
1	Harry Potter	Super kniha...	350	2004	UPDATE, DELETE
2	Pan Prstenov	Dobra kniha...	450	1890	UPDATE, DELETE

[1,2,3 << >> LAST]

Laravel zadanie c.2

Aplikáciu je potrebné doplniť o ďalšiu funkčnosť a to pridanie žanru, do ktorého kniha patrí.

V menu bude dostupná nová položka pod názvom "Genre", kde bude podobné rozhranie na správu žanrov – rovnako ako ste vytvorili pre správu kníh.

Následujúcim krokom bude potrebné prepojiť žanre s knihami a to spôsobom, že kniha môže mať len jeden žanr a žanr môže byť priradený k viacerým kniham, tzn. Kardinalita 1:N

Postup:

1. Vytvoríte CRUD administráciu pre žanre podľa vzoru zadania c.1 (Formu nechávame na riešiteľovi)
2. Do listingu žanrov do tabuľky pridať stĺpec "Počet kníh", kde sa zobrazí počet kníh daného žanru
3. Do tabuľky/datagridu kníh zo zadania c.1 pridať stĺpec "Žanr", ktorý bude niešť hodnotu k akému žanru je kniha priradená (názov) a samozrejme doplniť výber príslušného žanru aj v kroku vytvárania a updatovania kníh. Tento údaj bude povinný a mal by pri jeho výbere fungovať autocomplete.

OTÁZKA:

Čo sa stane keď k nejakej knihe priradíte žanr a ten následne zmazáte? Ako tento prípad vyriešite? (netreba implementovať, len vysvetliť).

Laravel zadanie c.3

Vytvoríte middleware, ktorý bude suplovať funkciu "monitoringu".

1. Vytvoríte klasický laravel middleware, na ktorý sa napoja všetky routy aplikácie
2. Middleware bude kontrolovať, či aplikácia nehodila exception, respektíve niekde nevrátila chybový stav (50x) error
3. Ak middleware nejakú chybu zachytí, zaloguje ju a pošle email na emailovú adresu z konfigurácie (.env premennej)
4. Vytvoríte routu /simulate/error, kde táto ruta buď vyhodí exception alebo nasimuluje iný druh erroru (aby sa to dalo otestovať)

Zadanie c.4

Vytvoríte jednoduchý bash script, ktorý spustí všetky príkazy potrebné k rozbehnutiu projektu priamo z Git repozitára a naseeduje databázu nejakými dummy dátami.

Zadanie c.5

Databázu nevytvárate cez laravel migrácie, ale ručne cez DDL príkazy, spolu aj s referenčnými integritami. Ukázať na pár záznamoch aj použitie DML príkazov na naplnenie databázy.