DELTA – Střední škola informatiky a ekonomie s.r.o.

Ke Kamenci 151, PARDUBICE

**IS – LPU Pardubice**

Informační systém – OK LOKOMOTIVA PARDUBICE

Zeman Martin

4.A

Obor č. 18: Informatika

2021/22

**Zadání maturitního projektu z informatických předmětů**

Jméno a příjmení: *Martin Zeman*

Školní rok: *2021/2022*

Třída: *3.A*

Obor: *Informační technologie 18-20-M/01*

Téma práce: *Modernizace rozhraní IS LPU*

Vedoucí práce: *RNDr. Jan Koupil, Ph.D.*

**Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:**

Cílem práce je vytvořit nové uživatelské rozhraní stávajícího informačního systému oddílu orientačního běhu LPU Pardubice.

Webové uživatelské rozhraní bude navrženo jako responzivní a bude se připojovat ke stávající databázi MySQL.

V případě potřeby autor provede i modernizaci databáze a navrhne úpravy původního administračního rozhraní IS tak aby zůstalo funkční.

**Stručný časový harmonogram (s daty a konkretizovanými úkoly):**

• září: Analýza stávajícího řešení, hledání nekompatibility s aktuálními verzemi databáze

• říjen: migrace dat na aktuální verzi databáze

• listopad-prosinec: Implementace vlastního rozhraní

• leden-únor: Testování, opravy chyb, připojení původní administrace

**Prohlášení**

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze zdroje a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 30.3.2022

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(vlastnoruční podpis)

**Poděkování**

Největší díky patří vedoucímu mého maturitního projektu RNDr. Janu Koupilovi, Ph.D. za odborné rady a nápady, které mi pomohly s tímto projektem.

A druhé díky patří mým spolužákům, kteří mě podrželi a povzbuzovali, i když projekt nešel zdaleka podle představ.

**Resumé**

Cílem projektu je vytvořit nové uživatelské rozhraní pro členy pardubického klubu OK LOKOMOTIVA.

Uživatelé zde budou mít přehled o novinkách v oddíle, plánovaných závodech na tuto sezónu a dalších akcích, které si pro ně klub připraví.

Přímo z tohoto systému se také uživatelé mohou na jednotlivé akce registrovat nebo se z nich naopak odhlásit.

Administrátoři databáze mají svojí vlastní stránku pro administraci, která nijak nesouvisí s tímto projektem

**Klíčová slova**

Webová aplikace, IS – LPU, Laravel, MySQL, PHP, Blade, Orientační běh, Maturitní práce

**Resume**

The main goal of this project is to create a user interface for the members of Pardubice sports club OK LOKOMOTIVA.

The interface will provide a way for it‘s users to view a news summary about their troop, planned races for this season, and other events organized by the club. They will also be able to sign up or sign out of the club activities.

Database administrators will have their own page for administration which is unrelated to this project.

**Keywords**

Web application, IS – LPU, Laravel, MySQL, PHP, Blade, Orienteering, Graduation thesis

**Obsah**

Obsah

[1 Úvod 13](#_Toc98139817)

[2 Použité technologie 13](#_Toc98139818)

[2.1 Framework 13](#_Toc98139819)

[2.1.1 Laravel 13](#_Toc98139820)

[2.2 Šablonovací engine 14](#_Toc98139821)

[2.2.1 Blade 14](#_Toc98139822)

[2.3 ORM 14](#_Toc98139823)

[2.3.1 Eloquent 14](#_Toc98139824)

[2.4 Databáze 14](#_Toc98139825)

[2.4.1 MySQL 15](#_Toc98139826)

[2.4.2 Normalizace databáze 15](#_Toc98139827)

[3 Praktická část 18](#_Toc98139828)

[3.1 Databáze 18](#_Toc98139829)

[3.1.1 Tabulka admin 19](#_Toc98139830)

[3.1.2 Tabulka akce 20](#_Toc98139831)

[3.1.3 Tabulka akce\_komentar 20](#_Toc98139832)

[3.1.4 Tabulka akce\_prihlaska 21](#_Toc98139833)

[3.1.5 Tabulka akce\_prihlasky\_def 21](#_Toc98139834)

[3.1.6 Tabulka aktuality 21](#_Toc98139835)

[3.1.7 Tabulka bazar 22](#_Toc98139836)

[3.1.8 Tabulka bzsoubor 22](#_Toc98139837)

[3.1.9 Tabulka chattr 23](#_Toc98139838)

[3.1.10 Tabulka diskuze 23](#_Toc98139839)

[3.1.11 Tabulka login\_check 23](#_Toc98139840)

[3.1.12 Tabulka obleceni 24](#_Toc98139841)

[3.1.13 Tabulka poradani 24](#_Toc98139842)

[3.1.14 Tabulka prihlasky 25](#_Toc98139843)

[3.1.15 Tabulka prihlaskyvic 26](#_Toc98139844)

[3.1.16 Tabulka soutez\_odpovedi 27](#_Toc98139845)

[3.1.17 Tabulka soutez\_otazky 28](#_Toc98139846)

[3.1.18 Tabulka termin 28](#_Toc98139847)

[3.1.19 Tabulka tmakce 29](#_Toc98139848)

[3.1.20 Tabulka tmsoubor 29](#_Toc98139849)

[3.1.21 Tabulka uzivatele 30](#_Toc98139850)

[3.1.22 Tabulka vklady 30](#_Toc98139851)

[3.1.23 Tabulka zavody 31](#_Toc98139852)

[3.1.24 Tabulka zavody\_kategorie 32](#_Toc98139853)

[3.1.25 Tabulka zavvic 33](#_Toc98139854)

[3.1.26 Tabulka termin\_zavvic 34](#_Toc98139855)

[3.1.27 Tabulka zavvic\_termin\_prihlasek 34](#_Toc98139856)

[3.1.28 Tabulka zebprihl 34](#_Toc98139857)

[3.1.29 Tabulka zebricek 35](#_Toc98139858)

# Úvod

Projekt by měl nahradit stávající uživatelské rozhraní pardubického oddílu orientačního běhu OK LOKOMOTVA PARDUBICE. Staré rozhraní bylo neresponzivní, mělo zastaralý design a díky tomu nebylo uživatelsky přívětivé.

Tento projekt má za cíl tyto neduhy vymýtit, a vytvořit hezké a uživatelsky přívětivé rozhraní, ať už se na něj koukáte na počítači nebo telefonu.

Dále je cílem zmodernizovat a zefektivnit stávající databázi. Ta v současném stavu běží na zastaralé verzi MySQL a chybí v ní jakékoliv hlídání integrity databáze například v podobě cizích klíčů.

# Použité technologie

## Framework

Framework je software, který slouží jako základna, na která se staví následná konkrétní aplikace.

Každý framework je vytvořený k jinému účelu a má podle toho i předpřipravené funkce, které pomohou programátorovi zpřehlednit a zjednodušit jeho práci a kód samotný. Každý programátor však musí zvolit správný framework pro to, co po aplikaci požaduje, aby dělala.

### Laravel

Laravel je open source framework pro webové aplikace pod licencí MIT a postavený na jazyce PHP. Je založený na architektuře MVC (model-view-controler). Což znamená, že kód je rozdělený do třech základních kategorií. Do modelu, který obsahuje data a funkce, view, to slouží k prezentaci dat například v HTML, nebo v případě Laravelu Blade, a controleru, který spravuje komunikace mezi modelem, view a uživatelem.

Laravel je designovaný hlavně pro komunikaci s MySQL databází. Ale dá se použít i s dalšími druhy SQL databází, a dokonce i s NoSQL. Celkově je připravený na reálné použití a má připravenou většinu věcí které jsou společné pro většinu projektů, například registraci a přihlašování uživatelů. Má pro to připravené jak funkce, tak i tabulky, které vám vytvoří jedním příkazem hned po tom, co databázi připojíte k aplikaci.

[[3]](#_Zdroje)

## Šablonovací engine

Šablonovací engine je software, který se stará o prezenční vrstvu databáze, o takzvaný frontend. Kód v něm zapsaný je pak pro vyhledávače překládán do čistého HTML. Proč však nepoužíváme čisté HTML? Důvod je jednoduchý. Šablonovací engine nám dává možnost používat například podmínky nebo cykly, což výrazně zpřehledňuje výsledný kód.

### Blade

Blade je jednoduchý, avšak mocný šablonovací engine, který je součástí Laravelu. Tento šablonovací engine má svojí hlavní výhodu v tom, že nám opravdu pouze zpřehledňuje kód, zatímco vše, co je v něm napsané, je uložené ještě v druhém PHP souboru, kde už je to opravdu pouze čisté PHP, které se přepisuje pouze v případě změny zdrojového kódu, tedy souboru Blade. Tento způsob zajišťuje co nejmenší zátěž výpočetního výkonu serveru, na kterém nám aplikace běží.

Soubory je třeba pojmenovávat s koncovkou .blade.php a v Laravelu jsou ukládány do adresáře /resources/views.

[[5]](#_Zdroje)

## ORM

Orm neboli Object Relational Mapping je vrstva mezi databází a samotným programem. Funkcí této vrstvy je vytáhnout data z databáze, a namapovat je na objekt. To umožňuje jednoduší a přehlednější práci s databází.

[[2]](#_Zdroje)

### Eloquent

Eloquent je ORM rozhraní které používá Laravel. Ten nám zde zajišťuje veškerou komunikaci s databází, jak co se týče dotazů na data, tak i jejich importování případně upravování. A pomocí takzvaných migrací zde můžeme i vytvářet a upravovat tabulky.

[[4]](#_Zdroje)

## Databáze

Pokud začneme historií databází, tak nejstarší předchůdce databází jsou papírové kartotéky. Ty fungovaly na stejném principu jako fungují dnešní databáze. Data zde byla členěna podle jednotlivých kritérií a mohly se na sebe navzájem odkazovat, například pacient má tuto nemoc, informace o nemoci jsou uloženy v této části kartotéky v tomto svazku.

Časem se data přesunula z papíru na výpočetní techniku. Nejdříve to byly děrné štítky a následně, jak vývoj techniky pokračoval, se data přesouvala na výkonnější a efektivnější úložná média. Postupně vznikly hierarchické databáze, poté relační databáze, následně objektové, avšak ty nedostály svým cílům, a místo aby vytlačily relační, tak se s nimi spojily, a tak vznikly dnešní objektově-relační databáze.

Mezi databázové objekty patří například pohledy, indexy, procedury a funkce nebo E-R schéma

### MySQL

MySQL je databázový systém založený na relačním konceptu databáze s podporou objektů. Systém je multiplatformní. Komunikace s databází probíhá v jazyce SQL, který je však drobně modifikovaný. Není tedy možné vzít příkaz z této databáze a vložit ho například do PostgeSQL (jiný databázový systém založený také na jazyce SQL).

MySQL je v dnešní době hojně využívaná pro drobné projekty, a to hlavně díky jednoduchosti implementace. Druhým zásadním důvodem je dostupnost. MySQL je kompletně zdarma. Velice často je využívána v kombinaci s backendem, který je napsaný v jazyce PHP.

MySQL je jednou z jednodušších SQL databází, ale to má vliv na výkon. Databáze je designovaná hlavně na rychlost.

[[6]](#_Zdroje)

### Normalizace databáze

Normalizace databáze je postup, při kterém přeorganizujeme, roztřídíme a rozdělíme data tak, abychom co nejefektivněji využívali potenciál relačních databází.

Při normalizaci databáze řešíme konzistenci (úplnost) a redundanci (opakování) dat.

Při normalizaci databáze se postupuje podle normálových forem. Normálové formy jsou pravidla, podle kterých upravujeme strukturu databáze. Při normalizaci musíme splňovat normálové formy postupně jelikož pro splnění každé normálové formy je vyžadováno splnění všech předchozích forem.

Ve většině projektů se požaduje splnění 3NF a o další se nesnažíme. Je to způsobeno tím, že výsledek ve výkonu a udržitelnosti databáze neodpovídá vynaloženému úsilí, které je třeba, aby databáze tyto normálové formy splňovala.

#### Nenormalizovaná forma (UNF)

Nenormalizovaná forma byla zavedena pro databáze nesplňující normálové formy.

Pro její splnění stačí, že databáze existuje.

#### Nultá normální forma (0NF)

V nulté normálové formě se tabulka vyskytne tehdy, jsou-li nějaká data uložená v jednom sloupečku tabulky dále dělitelná. Což je však přímo v rozporu s relačním modelem, který nám říká, že data v jednom poli jsou dále nedělitelná.

Tabulka splňuje buďto nultou, nebo první normálovou formu, jelikož tyto dvě si navzájem odporují.

#### První normální forma (1NF)

První normálová forma nám naopak říká, že každé pole tabulky obsahuje atomické, dále nedělitelné, hodnoty.

#### Druhá normální forma (2NF)

Druhá normálová forma nám říká, že každý neklíčový atribut je plně závislý na každém kandidátním klíči.

Jinak řečeno každý atribut, který není kandidátním klíčem tabulky, musí souviset se všemi kandidátními klíči tabulky. Pokud nesouvisí tak by měl být umístěn do jiné tabulky, kde toto pravidlo splňovat bude.

#### Třetí normální forma (3NF)

Třetí normálová forma nám říká, že všechny neklíčové atributy musí být vzájemně nezávislé.

Tedy pokud odstraníme kterýkoliv sloupeček neklíčových hodnot tabulky, neporuší se nám tím integrita dat uložených v této tabulce.

#### Boyce-Coddova normální forma (BCNF)

Boyce-Coddova normální forma nám říká že, atributy, které jsou součásti primárního klíče, musí být vzájemně nezávislé.

Tedy žádný sloupeček, který je primárním klíčem, nebo jeho součástí, nesmí být závislí na jakémkoli jiném sloupečku tabulky.

#### Čtvrtá normální forma (4NF)

Čtvrtá normálová forma nám říká, že relace popisuje pouze příčinnou souvislost mezi klíčem a atributy.

To znamená, že v jedné tabulce máme uloženy pouze atributy, které spolu souvisí. Každý atribut tedy musí souviset se všemi ostatními atributy tabulky. Nestačí, že všechny atributy souvisí s primárním klíčem.

#### Pátá normální forma (5NF)

Pátá normálová forma nám říká, že relace již není možno bezeztrátově rozložit.

[[1]](#_Zdroje)

# Praktická část

## Databáze

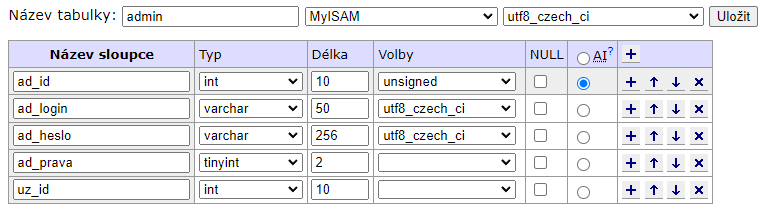
Jak jsem již psal v části technologií, pro tento projekt jsem použil SQL databázi. Konkrétně je to databáze MySQL. Výběr nebyl složitý, jelikož v MySQL byla vytvořena i původní databáze, kterou jsem dostal jako jeden z podkladů k tomuto projektu. Naštěstí MySQL je databáze, která naprosto vyhovuje veškerým potřebám tohoto projektu. Je jednoduchá a rychlá, což je vše, co na tento projekt potřebujeme. Navíc není problém si tuto databázi rozjet zdarma na jakémkoliv stroji.

Co se však týče konkrétní databáze, kterou jsem dostal jako podklad pro projekt, tak to už je něco úplně jiného. Většina tabulek byla nevyhovující, a to ať se bavíme o použité kolekci znaků, nebo například o vazbách mezi jednotlivými tabulkami.

Databáze také běžela na několik let staré verzi, a měla implementované funkce, které již nové verze MySQL nepodporují.

Díky tomuto všemu musel být první krok projektu úprava stávající databáze.

Obrázek Detail tabulky admin



Na obrázku výše vidíte původní tabulku admin. Na obrázku je vidět hned několik chyb, které se nám objevují ve skoro každé tabulce databáze, některá dokonce v každé.

V prvním kroku musela být změněna znaková kolekce z utf8\_czech\_ci na utf8mb4\_czech\_ci a to jak pro celou tabulku, tak i pro všechny sloupce typu varchar. Tato změna byla nutná, aby nám v databázi nedělali problém čtyřbitové znaky jako mohou být například emotikony.

Zároveň s tím se také měnil formát ukládání dat v tabulkách z formátu MyISAM na formát InnoDB. To je pro chod databáze zásadní změna. Formát InnoDB je sice náročnější na výkon databáze, ale podporuje cizí klíče, což je základ pro udržení konzistence dat v SQL databázích. A v původní databázi nebyly cizí klíče vůbec implementovány.

Po tomto kroku tedy bylo zapotřebí projít celou databázi a zjistit všechny vazby mezi tabulkami. Po této analýze museli být všechny tyto vazby definovány pomocí cizích klíčů.

Obrázek Schéma databáze

Obsah obrázku text, interiér, černá

Popis byl vytvořen automatickyNa obrázku víše vidíte všechny nadefinované relace mezi tabulkami. Pravidla pro tyto cizí klíče byly po konzultaci s klientem nastaveny následovně: Při změně dat, která by zapříčinila nekonzistenci dat, nám databáze tuto změnu zablokuje a neprovede se a při mazání dat budou kaskádovitě smazána i všechna závislá data. Díky tomu se nám nestane, že bychom někde měli cizí klíč, který by odkazoval na neexistující řádek tabulky.

Dalším krokem bylo upravit jednotlivé tabulky tak aby splňovali alespoň třetí normální formu. U tohoto kroku jsem však narazil na neshody s klientem, kde podle normálních forem a obecně známých pravidel měla tabulka vypadat jedním stylem, ale klient si nepřál tak rozsáhlé změny v databázi a tím pádem jsme museli najít nějaký kompromis.

### Tabulka admin

Obrázek Tabulka admin

Obsah obrázku text

Popis byl vytvořen automaticky

Z této tabulky byli odebrány dva sloupce, těmi jsou ad\_login a ad\_heslo, ve kterých byla uložena stejná data jako v tabulce uživatelé, na kterou je každý záznam z této tabulky pomocí cizího klíče navázán.

Také byl přidán samotný cizí klíč na sloupeček uz\_id, který odkazuje, jak už bylo zmíněno na sloupeček uz\_id z tabulky uzivatelé.

### Tabulka akce

Obrázek Tabulka akce

Obsah obrázku stůl

Popis byl vytvořen automaticky

V této tabulce nebyly potřeba až tak velké změny. Na sloupeček ak\_spravce byl nadefinován cizí klíč odkazující na sloupeček uz\_id z tabulky uživatelé. A ze sloupců s typem date byla odebrána vlastnost nullable, protože akce musí mít datum začátku i konce.

### Tabulka akce\_komentar

Obrázek Tabulka akce\_komentar

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

V této tabulce byla přidána defaultní hodnota na sloupeček ako\_datumcas. A dále byli přidány cizí klíče na sloupečky uz\_id a ak\_id. Uz\_id odkazuje na tabulku uživatelé a ak\_id odkazuje na tabulku akce.

### Tabulka akce\_prihlaska

Obrázek Tabulka akce\_prihlaska

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

Do této tabulky musel být přidán sloupeček pr\_id, jelikož tabulka původně neměla žádný primární klíč a nebyl zde nadefinovaný ani složený klíč. Dále se zde přidala defaultní hodnota na sloupeček pr\_datum\_prihlaseni a cizí klíče na sloupečky uz\_id a ak\_id, která jsou stejné jako v předchozí tabulce.

### Tabulka akce\_prihlasky\_def

Obrázek Tabulka akce\_prihlasky\_def

Obsah obrázku text

Popis byl vytvořen automaticky

V této tabulce byl přejmenován sloupeček apd\_id z původního zavádějícího názvu a byl na něj přidán index primárního klíče. Dále byl přidán sloupeček ak\_id který se nastavil jako cizí klíč odkazující na tabulku akce.

### Tabulka aktuality

Obrázek Tabulka aktuality

Obsah obrázku stůl

Popis byl vytvořen automaticky

Na této tabulce nebyli prováděny žádné další úpravy, krom již víše zmíněných.

### Tabulka bazar

Obrázek Tabulka bazar

Obsah obrázku stůl

Popis byl vytvořen automaticky

V této tabulce byla na sloupeček bz\_datum přidána defaultní hodnota a na sloupeček uz\_id byl přidán cizí klíč odkazující na tabulku uživatelé.

### Tabulka bzsoubor

Obrázek Tabulka bzsoubor

Obsah obrázku stůl

Popis byl vytvořen automaticky

V této tabulce bylo potřeba pouze přidat cizí klíč na sloupeček bz\_id odkazující na tabulku bazar.

### Tabulka chattr

Obrázek Tabulka chattr

Obsah obrázku text

Popis byl vytvořen automaticky

V této tabulce se odebrala vlastnost nullable ze sloupečků tr\_datum a tr\_odkaz a na sloupeček tr\_datum byla zároveň přidána defaultní hodnota. Dále na sloupeček tr\_autor byl přidán cizí klíč odkazující na tabulku uživatelé.

### Tabulka diskuze

Obrázek Tabulka diskuze

Obsah obrázku text

Popis byl vytvořen automaticky

V táto tabulce byla opět vlastnost nullable na sloupečku di\_datum nahrazena defaultní hodnotou. A na sloupec di\_autor byl přidán cizí klíč.

### Tabulka login\_check

Obrázek Tabulka login\_check

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

V této tabulce byl pouze přidán cizí klíč na sloupeček lc\_uzid odkazující na tabulku uživatelé.

### Tabulka obleceni

Obrázek Tabulka obleceni

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

V této tabulce byl pouze přidán cizí klíč na sloupeček uz\_id odkazující na tabulku uživatelé.

### Tabulka poradani

Obrázek Tabulka poradani

Obsah obrázku stůl

Popis byl vytvořen automaticky

Z této tabulky byla odebrána defaultní hodnota ze sloupečku uz\_id, která vkládala 0, a zároveň na něj byl přidán cizí klíč odkazující na tabulku uživatelé.

### Tabulka prihlasky

Obrázek Tabulka prihlasky

Obsah obrázku stůl

Popis byl vytvořen automaticky

V této tabulce bylo potřeba přidat sloupeček pr\_datum\_admin, jelikož admin původně, když upravil tabulku, přepsal sloupeček pr\_datum\_prihlaseni, čímž jsme přišli o data kdy se přihlásil na závod uživatel.

Dále zde byli přidány cizí klíče na sloupečky uz\_id a za\_id odkazující na tabulky uživatelé a závody.

Přidala se zde defaultní hodnota na sloupeček pr\_datum\_prihlaseni a byli odebrány defaultní hodnoty ze sloupců uz\_id a za\_id.

A přidal se sloupec pr\_id který slouží jako primární klíč.

Sloupeček pr\_datum by mohl být odstraněn, protože je v něm stejná informace jako v tabulce závody, ale klient ho tam chtěl nechat kvůli řazení.

Do sloupečku pr\_kategorie by se nově mělo správně ukládat id kategorie a nastavit cizí klíč, ale klientovy se toto řešení zdálo nepřehledné, proto se zanechalo ukládání varcharové hodnoty kategorie.

### Tabulka prihlaskyvic

Obrázek Tabulka prihlaskyvic

Obsah obrázku stůl

Popis byl vytvořen automaticky

Do této tabulky byl přidán sloupeček pr\_id a nastaven jako primární identifikátor.

Na sloupeček pr\_datumprihl byla přidána defaultní hodnota a odstraněna původní vlastnost nullable.

Sloupeček pr\_kategorie na tom je problematicky stejně jak to je již vysvětleno víše u jednodenních přihlášek.

### Tabulka soutez\_odpovedi

Obrázek Tabulka soutez\_odpovedi

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

Do této tabulky byl přidán sloupeček sod\_id a byl nastaven jako primární klíč.

Dále sloupečku sod\_datum byla odebrána vlastnost nullable a byla přidána defaultní hodnota.

A na sloupečky sot\_id a uz\_id byly přidány cizí klíče odkazující na tabulky uživatelé a soutez\_otazky.

### Tabulka soutez\_otazky

Obrázek Tabulka soutez\_otazky

Obsah obrázku stůl

Popis byl vytvořen automaticky

V této tabulce byla ze sloupců sot\_start a sot\_end odstraněna vlastnost nullable. A na sloupeček uz\_id byl přidán cizí klíč odkazující na tabulku uživatelé.

### Tabulka termin

Obrázek Tabulka termin

Obsah obrázku text

Popis byl vytvořen automaticky

Tato tabulka je na uložení pouze jedné hodnoty. Celá tabulka by měla být předělána na tabulku konstant, protože toto je jediná konstanta v databázi a ostatní konstanty jsou uloženy v textovém souboru v aplikaci, ale klient si takové změny nepřeje, jelikož by si musel předělávat administrátorské rozhraní.

### Tabulka tmakce

Obrázek Tabulka tmakce

Obsah obrázku text

Popis byl vytvořen automaticky

Do této tabulky byl pouze přidán cizí klíč na sloupeček uz\_id odkazující na tabulku uživatelé a byla z něho odebrána defaultní hodnota 0.

### Tabulka tmsoubor

Obrázek Tabulka tmsoubor

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

Do této tabulky byl pouze přidán cizí klíč na sloupeček tm\_id odkazující na tabulku tmakce a byla z něho odebrána defaultní hodnota 0.

### Tabulka uzivatele

Obrázek Tabulka uzivatele

Obsah obrázku stůl

Popis byl vytvořen automaticky

Na této tabulce nebyli provedeny žádné změny.

### Tabulka vklady

Obrázek Tabulka vklady

Obsah obrázku text

Popis byl vytvořen automaticky

V této tabulce byla ze sloupečku vk\_termin odebrána vlastnost nullable. A na sloupeček uz\_id byl přidán cizí klíč odkazující na tabulku uživatelé.

### Tabulka zavody

Obrázek Tabulka zavody

Obsah obrázku stůl

Popis byl vytvořen automaticky

Z této tabulky vylo odebráno několik sloupečků, které deklarovali kategorie a vklady pro dané kategorie, a tyto sloupečky byli nahrazeny novou tabulkou zavody\_kategorie. Původní řešení bylo osm sloupečků pro kategorie a osm pro vklady k těmto kategoriím, ale problém byl, že na každém závodě byl jiný počet kategorií, a tím pádem nebyli většinou všechny tyto sloupečky využity. Proto se nahradili novou tabulkou kde je nyní jedno kolik kategorií který závod má jelikož jsou na sobě v tomto ohledu závody naprosto nezávislé.

Dále zde byla z datumů odebrána vlastnost nullable, protože každý závod musí mít začátek a konec.

### Tabulka zavody\_kategorie

Obrázek Tabulka zavody\_kategorie

Obsah obrázku stůl

Popis byl vytvořen automaticky

Toto je nová tabulka vzniklá z dříve popsaných důvodů.

### Tabulka zavvic

Obrázek Tabulka zavvic

Obsah obrázku stůl

Popis byl vytvořen automaticky

Z této tabulky muselo být také odstraněno několik sloupců, a to ze stejného důvodu jako u tabulky závody. Tyto sloupce byli nahrazeny dvěma tabulkami, a to termin\_zavvic a zavvic\_termin\_prihlasek. V tabulce termin\_zavvic jsou uloženy datumy a tagy jednotlivých datumů které nám určují kolikáté kolo přihlášek je tento termín. To by šlo sice zjišťovat i automaticky, ale zatěžovalo by to výpočetní výkon a klient si také chce mezi těmito koly sám ručně přepínat. To v kolikátém kole právě přihlášky jsou se určuje podle sloupečku za\_typ\_kateg, který si klient chce sám upravovat. Z tohoto sloupečku poté vychází druhá tabulka, ve které jsou deklarovány kategorie, na které se může uživatel přihlásit, a ceny jednotlivých kategorií podle toho ve kterém kole termínů se přihlásíte. U těchto závodů to funguje tak, že čím dříve se přihlásíte, tím méně vás to bude stát.

### Tabulka termin\_zavvic

Obrázek Tabulka termin\_zevvic

Obsah obrázku stůl

Popis byl vytvořen automaticky

Toto je nová tabulka vzniklá z dříve popsaných důvodů.

### Tabulka zavvic\_termin\_prihlasek

Obrázek Tabulka zavvic\_termin\_prihlasek

Obsah obrázku text, stůl

Popis byl vytvořen automaticky

Toto je nová tabulka vzniklá z dříve popsaných důvodů.

### Tabulka zebprihl

Obrázek Tabulka zebprihl

Obsah obrázku stůl

Popis byl vytvořen automaticky

Do této tabulky byl přidán sloupeček zp\_id a byl nastaven jako primární klíč.

Ze sloupečků uz\_id a ze\_id byla odebrána vlastnost nullable, a naopak byl na tyto sloupečky přidán cizí klíč odkazující na tabulky uživatelé a žebříček.

A sloupeček zp\_prijmeni byl odebrán, protože přímení je uloženo právě v tabulce uživatelé.

### Tabulka zebricek

Obrázek Tabulka zebricek

Obsah obrázku text

Popis byl vytvořen automaticky

V této tabulce byla pouze odebrána ze sloupečku datum možnost nullable.

## Webová aplikace

Jak je zmíněno víše, webová aplikace je postavená na PHP frameworku Laravel. A ten se skládá z několika částí. Pro práci s uživateli byl použit middleware od Laravelu, který byl upraven pro potřeby této aplikace. Dále se je zde router, který převezme informace ze zadané adresy a předá je příslušné funkci v kontroleru. Kontroler je následně zpracuje, provede požadované akce, které v sobě má napsané, a předá výsledek view vrstvě. Ta výsledek zobrazí uživateli, dokud ten neprovede další akci a celé se to znovu nezopakuje.

### Migrace

Migrace jsou soubory, které používá Laravel pro práci se strukturou databáze. Jsou uloženy ve složce *database/migrations* a píší se sem veškeré změny struktury databáze. I v tomto projektu máme každou tabulku definovanou ve vlastní migraci a poté je zde jedna velká migrace přidávající cizí klíče. To nám dohromady zajišťuje velice jednoduchou implementaci projektu, kdy nám pro rozchození stačí v souboru .env přepsat potřebné informace pro připojení databáze a následně v příkazovém řádku projektu zadáme příkaz *php artisan migrate* a tím máme rozchozenou celou databázi.

### Modely

Modely v Laravelu slouží taktéž pro práci s databází, ale tentokrát se jedná o práci s daty. Každá tabulka má k sobě vytvořený odpovídající model, ve kterém je řečeno, jaké sloupečky daná tabulka má, co je primární klíč, do jakých sloupců můžeme zapisovat napřímo a do jaký pouze s pomocí metod které pro to připravíme a tak dále. Jsou zde také definovány relace mezi tabulkami. Tyto modely pak slouží Laravelu pro převod databázových záznamů na objekty, s nimiž je dále jednodušší práce.

### Middleware

Obrázek 32 Úprava poskytovatelů

Obsah obrázku text

Popis byl vytvořen automatickyPro potřeby této aplikace bylo potřeba upravit originální middleware, který je nastaven na laravelem předdefinovanou tabulku. To pro naše potřeby nebylo vhodné, jelikož jsme dostali tabulku uživatelů od klienta. Proto jsme museli v souboru *config/auth.php* upravit poskytovatele, jak je vidět na obrázku vpravo, a v tomtéž souboru jsme tohoto poskytovatele museli poskytnout funkci guards namísto starého poskytovatele.

### Router

Obrázek Web router

Obsah obrázku text, exteriér, plaketa

Popis byl vytvořen automaticky

V routeru máme definovány veškeré podadresy naší webové aplikace. Router nám tuto adresu vezme a přesměruje nás do kontroleru na odpovídající funkci. Pokud zde máme get metodu, anebo post metodu, tak zde můžeme pomocí regulárních výrazů vyfiltrovat data, které chceme znát, od těch, co by mohli náš kód poškodit. Dále se zde používají takzvané divoké karty, které se píší do složených závorek a router vezme cokoliv co zde uživatel nebo program zadá a předá to dál pod názvem dané divoké karty. Taktéž se to dá filtrovat pomocí funkce where a pomocí regulárního výrazu.

### Controlery

Pro účely tohoto projektu byli vytvořeny dva kontrolery. Jeden pro autentifikaci uživatelů a druhý pro logiku stránek. V kontroleru pro autentifikaci uživatelů jsou vytvořené funkce pro přihlášení, odhlášení a vytvoření uživatelů. Vytvoření, neboli registrace, byla však pro účely tohoto projektu zablokována, neboť si klient nepřeje, aby se uživatelé mohli zaregistrovat sami, ale chce aby je mohl vytvořit pouze on ze své vlastní administrátorské webové aplikace. Druhý kontroler v sobě má funkce pro jednotlivé stránky a provádí vytahování dat z databáze. Pokud je potřeba, tak naformátuje do požadované podoby a pošle je dál pohledu. Dále jsou zde funkce, které naopak berou informace z postu a ukládají je do databáze. Každá funkce v tomto kontroleru také začíná tím, že ověří, zdali je uživatel přihlášen.

### Pohledy

Pohledy nám zde pomáhají s psaním kódu pohledu stránky, který vidí uživatel. Tyto pohledy se píší v jazyce Blade. Ten nám umožňuje základní řízení toku kódu, jakým jsou podmínky a cykly. Pro ušetření výpočetního výkonu serveru se však tento kód pak překládá do čistého PHP. Také zde používáme dědičnost jako v objektově orientovaném programování. Proto máme pro účely tohoto projektu vytvořený jeden pohled, ve kterém máme nadefinovanou hlavičku a patičku stránky a ostatní pohledy od tohoto hlavního dědí. Používáme zde také dynamické doplňování z proměnných, které byli předány kontrolerem. Toto doplňování se píše pomocí dvou do sebe vnořených složených závorek. Toto všechno nám umožní psát velice univerzální kód. V tomto projektu se toto používá například u detailu závodu, kde je vytvořený jeden pohled pro všechny závody. Navbar je rozdělený podle toho, jestli je uživatel přihlášený nebo ne a podle toho má také dané možnosti kam se může dostat.

# Zdroje

1. KULHAN, Jakub. Normalizace relačních databází. *Programujte.com* [online]. 2008 [cit. 2022-03-14]. Dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/>
2. ORM (Object Relational Mapping): automatizuje záznam dat. *hwlibre.com* [online]. [cit. 2022-03-14]. Dostupné z: <https://www.hwlibre.com/cs/orm-object-relational-mapping/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+hwlibreweb+%28Hardware+libre%29>
3. Laravel docs. *Laravel.com* [online]. [cit. 2022-03-14]. Dostupné z: <https://laravel.com/docs/9.x>
4. Laravel docs.: Eloquent. *Laravel.com* [online]. [cit. 2022-03-14]. Dostupné z: <https://laravel.com/docs/9.x/eloquent>
5. Laravel docs.: Blade. *Laravel.com* [online]. [cit. 2022-03-14]. Dostupné z: <https://laravel.com/docs/9.x/blade>
6. MySQL Documentation: What is MySQL?. *Mysql.com* [online]. [cit. 2022-03-14]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>