

DELTA – Střední škola informatiky a ekonomie,
s.r.o. Ke Kamenci 151, Pardubice

Webová aplikace pro rybáře

Jaroslav Pecka

4.A

Informační technologie (18-20-M/01)

2024/2025

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: Jaroslav Pecka

Školní rok: 2024/25

Třída: 4. A

Obor: Informační technologie 18-20-M/01

Téma práce: Webový portál a databáze se systémem rolí

Vedoucí práce: Jan Mottl

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

Cílem projektu bude vytvořit webový portál navázaný na databázi, který bude sloužit pro rybáře. V rámci webového portálu budou uživatelé spravovat skupiny a role v nich. Skupiny půjde zakládat soukromě, a přidání do nich bude možné pomocí pozvánky. Stejně tak budou fungovat veřejné skupiny, do kterých se bude moct přidat jakýkoliv uživatel s účtem na portálu. Ve skupinách bude možné zakládat komunikační vlákna a zaznamenávat své rybářské úlovky. Každý uživatel bude moct vytvářet katalog lokalit na rybaření a případně tyto lokality sdílet v rámci svých skupin. Ve skupinách pak půjde zobrazit zveřejněné lokality členů i úlovky ulovené na těchto lokalitách.

Harmonogram příprav

- Září: Rešerše a plánování řešení, seznámení s vývojářskými nástroji.
- Říjen: Implementace základních principů a nástrojů (přihlášení, propojení s databází), navrhnutí referenčního modelu databáze
- Listopad: Vložení funkcí pro přidání úlovků a lokality. Nastavení rolí v rámci databáze a portálu.
- Prosinec: Práce na dokončení hrubé podoby webu se všemi funkcionalitami.
- Leden: Grafická úprava webu.
- Únor: Testování webu na několika uživatelích. Ladění chyb a začátek práce na dokumentaci.
- Březen: Dokončení dokumentace a její odevzdání.

Prohlašuji, že jsem maturitní projekt vypracoval samostatně, výhradně s použitím uvedené literatury.

V Pardubicích, 18.3.2025

Poděkování

Rád bych poděkoval Mgr. Janu Mottlovi za vedení tohoto maturitního projektu, jeho cenné rady a poskytnuté konzultace. Také děkuji Bedřichu Kučerovi za pomoc s finálním nahostováním projektu.

Anotace

V této dokumentaci je popsána má webová aplikace pro rybář, její architektura a technologie mnou použité. V práci jsem také popisoval konkurenční aplikace a porovnal je s vlastním řešením. Aplikace je určena především pro organizaci a digitalizaci rybářských závodů, záznamu úlovků, lokalit a tvoření veřejných či soukromých skupin.

Klíčová slova

Rybaření, Laravel, Rybářské závody, PHP, JetStream, Správa skupin, Úlovky, SQLite

Annotation

This documentation describes my web application for anglers, its architecture, and the technologies I used. In my work, I also analyzed competing applications and compared them with my own solution. The application is primarily designed for organizing and digitizing fishing competitions, recording catches, locations, and creating public or private groups.

Keywords

Fishing, Laravel, Fishing competitions, PHP, JetStream, Groups, Catches, SQLite

Obsah

1. Úvod	7
1.1. Existující řešení	8
1.1.1. FishSurfing	8
1.1.2. Vranka	8
1.1.3. FishAngler	8
1.1.4. Fishing Statistics	9
2. Technologie	10
2.1. Laravel	10
2.2. Jetstream	10
2.3. Composer	10
2.4. NPM	11
2.5. SQLite	11
2.6. Tailwind	11
3. Popis aplikace	12
3.1. Skupiny	12
3.1.1. Založení nové skupiny	13
3.1.2. Správa skupiny	13
3.2. Příspěvky ve skupině	14
3.3. Lokality	14
3.3.1. Tvorba nové lokality	14
3.3.2. Detail lokality	15
3.4. Úlovky	15
3.4.1. Záznam úlovku	15
3.4.2. Detail úlovku	15
3.5. Závod	16
3.5.1. Nový závod	16
3.5.2. Správa závodu	17
3.5.3. Detail závodu	17
4. Use case diagram	18

5.	Architektura	21
5.1.	Základní koncept routování	22
5.2.	Uložiště	24
5.3.	Struktura databáze	27
5.4.	Ukládání souborů	29
6.	Reference.....	31
7.	Seznam obrázků	32

Úvod

Rybářství je v České republice velmi oblíbenou volnočasovou aktivitou, které se věnuje tisíce lidí. Moderní technologie se však v tomto oboru zatím příliš neprosadily, ačkoliv mohou značně usnadnit organizaci a evidenci důležitých údajů.

Jednou z oblastí, kde digitalizace chybí nejvíce, jsou rybářské závody. Většina organizátorů nadále používá papírovou evidenci závodníků, což způsobuje zdoluhavé vyhodnocování.

Cílem tohoto projektu je vytvořit webovou aplikaci určenou rybářům, která jim umožní sdílet své příspěvky ve skupinách, evidovat úlovky a zaznamenávat informace o rybářských revírech. Unikátní funkcionalitou aplikace je možnost organizace rybářských závodů, jejich správa a vyhodnocování v reálném čase. Na trhu v současnosti není k dispozici obdobná aplikace, která by tuto funkcionalitu nabízela.

Tento projekt tedy přináší inovativní řešení, které může rybářům i organizátorům závodů ušetřit čas, a zjednodušit vyhodnocování závodů.

1. Existující řešení

V této kapitole popíšu jednotlivé rybářské aplikace na trhu a v čem se popřípadě liší s mojí aplikací.

1.1. FishSurfing

Fishsurfing je unikátní celosvětová sociální síť pro rybáře, která umožňuje uživatelům sdílet své rybářské zážitky a spojovat se s dalšími nadšenci do rybaření, i když zrovna nejsou u vody. Aplikace nabízí řadu funkcí pro rybářskou komunitu, např. mapa revírů, fórum, stories, tržiště, fórum a bazar. (Fishsurfing, 2025)

Fishsurfing na rozdíl od mojí aplikace nepodporuje tvorbu skupin, organizací závodů, a je vhodný spíše pro jednotlivce než například místní organizaci.

1.2. Vranka

Vranka.cz byla oblíbená rybářská aplikace a webová stránka, která sloužila českým rybářům jako zdroj informací, diskusní platforma a prostor pro komunitu. Bohužel, v listopadu 2024 bylo oznámeno ukončení jejího provozu. Aplikace nabízela interaktivní mapy revírů, aktuální informace o podmínkách lovu, prostor pro diskuse, sumáře docházek, Atlas ryb a články. (Souček, 2024) (Vymětal, 2024)

Vranka nenabízela možnost skupin, ani organizací závodů. Byla určena pouze pro jednotlivce.

1.3. FishAngler

FishAngler je mobilní aplikace pro rybáře, která pomáhá objevovat nová rybářská místa, poskytuje předpovědi počasí a doporučuje nejlepší časy pro rybolov. Nabízí interaktivní mapy s potvrzenými úlovky, podvodními strukturami a vrstvami a skupiny. (FishAngler, 2025)

FishAngler na rozdíl od mé aplikace podporuje mapy s podvodními strukturami (pouze v placené verzi aplikace), ale nenabízí možnost organizace rybářských závodů.

1.4. Fishing Statistics

Velice nová aplikace pro profesionální organizaci rybářských závodů, vyvinuta především pro závod stairs2hell, aplikace nemá neplacenou verzi. Za použití na jednom závodě je cena 199€, při více závodech ročně je cena 159€/závod. Aplikace vypadá velice propracovaně a profesionálně, ale nehodí se pro menší závody, z důvodu její ceny. (European Carp Fishing Association, 2025)

Fishing Statistics je na rozdíl od mé aplikace pouze pro organizaci závodů a nemá žádnou neplacenou verzi.

2. Technologie

V této kapitole popisují nejdůležitější technologie, které můj projekt využívá

2.1. Laravel

Laravel je open-source PHP framework navržený pro vývoj moderních webových aplikací. Nabízí elegantní syntaxi, vestavěnou podporu pro autentizaci, a další užitečné funkce. Laravel využívá architekturu MVC (Model-View-Controller), což usnadňuje organizaci kódu a oddělení logiky aplikace od uživatelského rozhraní. Díky Eloquent ORM poskytuje efektivní práci s databází, a jeho podpora migrací usnadňuje spravování struktury databáze. Je to „páteř“ mého projektu. (Jalli, 2022) (Laravel, 2025)

2.2. Jetstream

Jetstream je oficiální startovací balíček pro Laravel, který poskytuje pokročilé funkce pro autentizaci uživatelů, správu týmů a dvoufázové ověřování. Nabízí předpřipravené komponenty pro registraci, přihlašování, resetování hesla a dvoufázové ověřování. (JetStream, 2025)

2.3. Composer

Composer je správce závislostí pro PHP, který umožňuje snadnou instalaci a správu balíčku a knihoven. Pomocí něj, lze jednoduše přidávat a aktualizovat rozšíření Laravelu, včetně balíčků třetích stran. Composer také umožňuje autoloading tříd, což zjednodušuje práci s kódem a minimalizuje potřebu manuálního importování souborů. (Adermann, a další, 2025)

2.4. NPM

Node Package Manager (NPM) je nástroj pro správu JavaScriptových balíčků. V Laravelu se využívá zejména pro frontendové technologie, jako je Tailwind CSS, Vue.js nebo další utility pro sestavování CSS a JavaScriptu. Pomocí NPM lze snadno spravovat a aktualizovat balíčky potřebné pro vývoj uživatelského rozhraní. (w3schools, 2025)

2.5. SQLite

SQLite je lehká relační databáze, která je integrovaná přímo do aplikace a nevyžaduje samostatný databázový server. V Laravelu se často používá při vývoji a testování aplikací díky své jednoduché konfiguraci a nenáročnosti na systémové prostředky. SQLite ukládá data do jediného souboru, což usnadňuje manipulaci s databází a přenos mezi různými prostředími. (Sqlite, 2025) (Wikipedia-Sqlite, 2025)

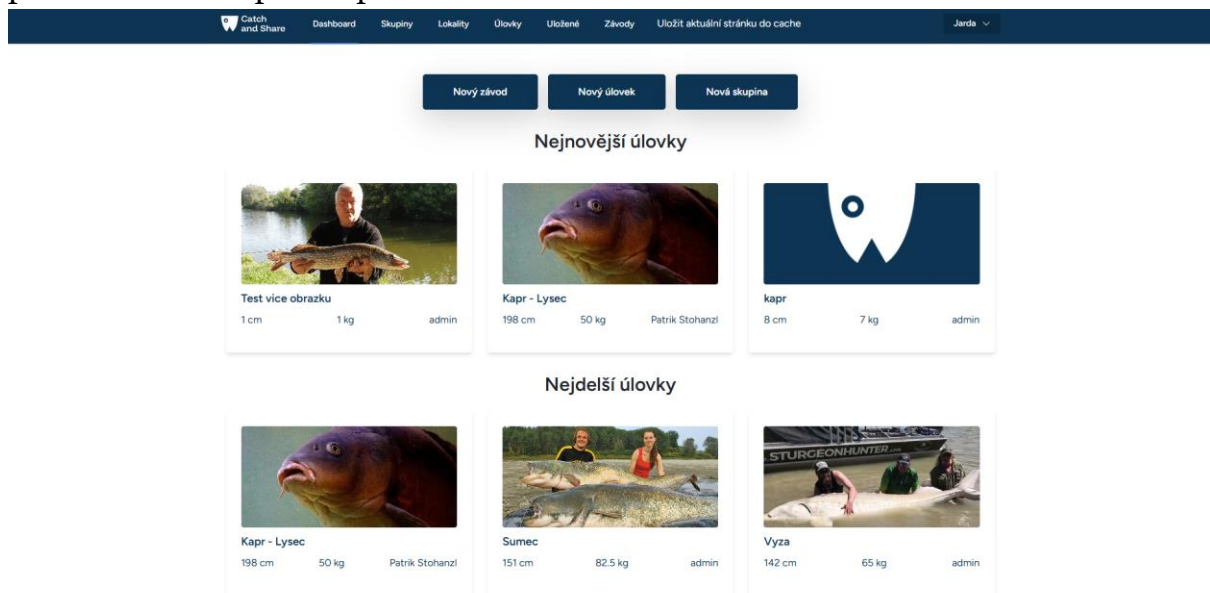
2.6. Tailwind

Tailwind CSS je utility-first CSS framework, který umožňuje rychlou a flexibilní tvorbu uživatelského rozhraní. Na rozdíl od tradičních CSS frameworků, které poskytují předdefinované komponenty, Tailwind umožňuje přímou úpravu stylů pomocí utility tříd. (Michálek, 2021)

3. Popis aplikace

V této části je shrnuto, co aplikace poskytuje, jak ji začít používat a popis jednotlivých funkcí, v následujících kapitolách budou věci více rozebrány po technické stránce.

Aplikace má welcome page, na ní je možnost se buď přihlásit, či registrovat. Bez přihlášení nelze aplikaci používat.



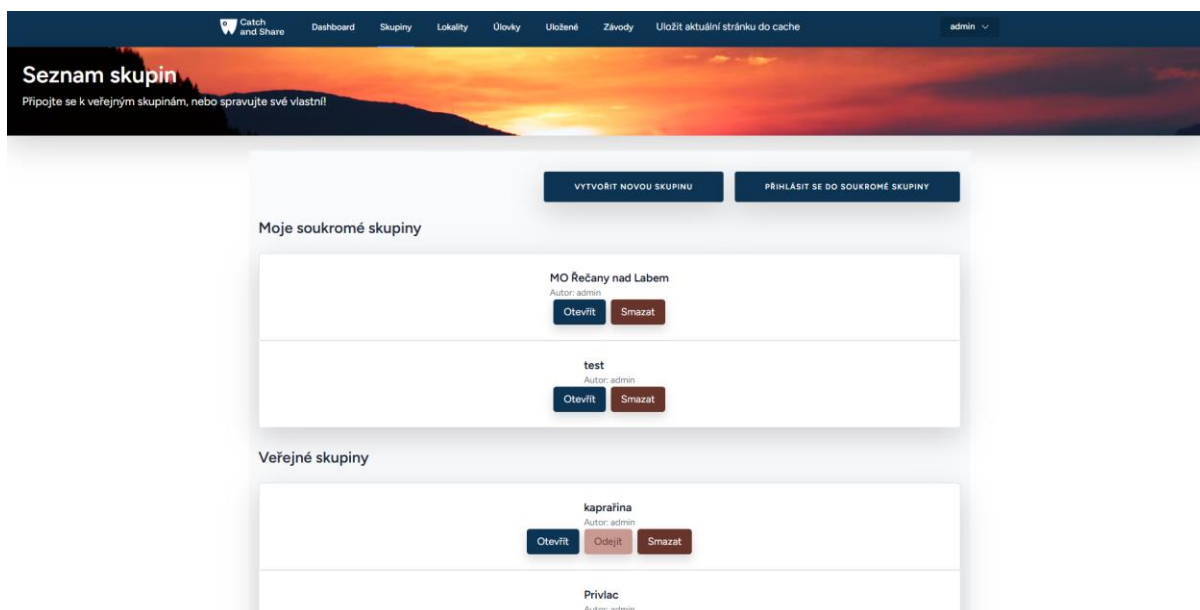
Obrázek 1 - Dashboard stránka [zdroj: autor]

Aplikace má 4 hlavní funkce:

- Skupiny
- Závody
- Úlovky
- Lokalita

3.1. Skupiny

Na stránce /skupiny jsou vypsány všechny veřejné skupiny a skupiny v nichž jsem členem. Dále jsou zde dvě tlačítka, "vytvořit novou skupinu" a "Přihlásit se do soukromé skupiny".



Obrázek 2 - Stránka skupiny [Zdroj: autor]

3.1.1. Založení nové skupiny

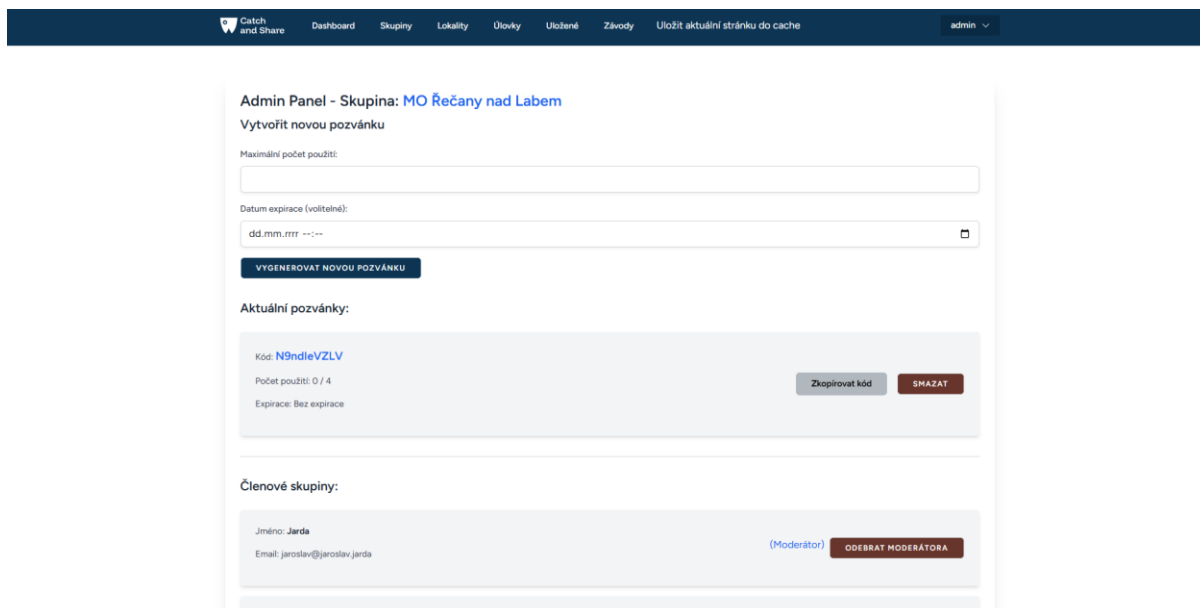
Po kliknutí na tlačítko “Vytvořit novou skupinu” se uživateli zobrazí jednoduchý formulář, v něm zadá název skupiny a poté vybere zda chce aby byla skupina soukromá či ne. Pokud uživatel bude chtít aby skupina byla soukromá, bude dále vyzván k tomu aby zadal heslo skupiny.

3.1.2. Správa skupiny

Po kliknutí tlačítka “Admin panel” má uživatel několik možností jak může spravovat skupinu.

Možnost je tvorba pozvánky: Admin může vygenerovat pozvánku pomocí níž se můžou do skupiny připojit ostatní uživatelé. Při tvorbě pozvánky si admin může upravit počet použití, nebo datum expirace pozvánky, tj může vygenerovat pozvánku, která bude platná pouze pro dva uživatele a pouze po dobu 2 dní. po vytvoření pozvánky se vygeneruje 10 místný kód, který může admin sdílet mezi osoby které chce aby se připojili, ti ten kód mohou zadat do formuláře na stránce skupiny pod tlačítkem “Připojit se do soukromé skupiny”.

Možnost je udělení Moderátora skupiny. Moderátor může ve skupině mazat příspěvky.



Obrázek 3 Admin panel skupiny [zdroj: autor]

3.2. Příspěvky ve skupině

Ve skupině je možnost zakládat příspěvky, kliknutím na tlačítko vytvořit nový příspěvek se uživateli zobrazí formulář do kterého zadá nadpis, text a popřípadě obrázky.

Každý uživatel může také komentovat příspěvky, a reagovat na jiné komentáře.

3.3. Lokality

Na stránce lokality je výpis všech soukromých + uživatelských lokalit. Na stránce si také může uživatel buď založit novou lokalitu, zobrazit si detail lokality, likenout lokalitu, či si lokalitu uložit.

3.3.1. Tvorba nové lokality

Po kliknutí tlačítka vytvořit lokalitu, se uživateli zobrazí formulář ve kterém zadá Název, druh a vybere na mapě kde se lokalita nachází.

Dále má také uživatel možnost vytvořit lokalitu soukromě, což udělá zakliknutím checkboxu “Lokalita je soukromá”. Po zakliknutí soukromosti si uživatel vybere zda chce aby byla lokalita soukromá pro skupinu jejíž je členem, nebo soukromá jen pro něj.

3.3.2. Detail lokality

Po kliknutí na detail lokality, se uživateli zobrazí informace o lokalitě, fotografie lokality, interaktivní mapa s vyznačenou lokalitou, a možnost přidat vlastní fotografii dané lokality.

3.4. Úlovky

Na stránce úlovky může uživatel vidět výpis všech veřejných úlovků + úlovky uživatele.

Dále může zaznamenat nový úlovek, zobrazit si detail úlovku, likeovat a ukládat úlovky.

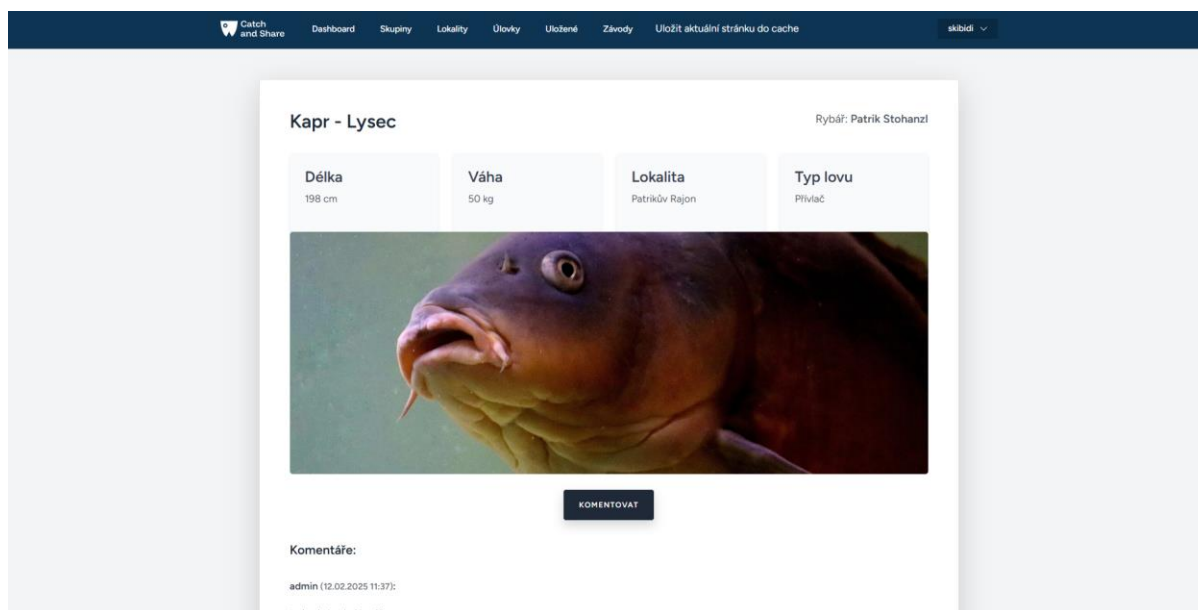
Výpis úlovku si může uživatel přizpůsobit několika způsoby, může si zobrazit jen své úlovky, může vyhledávat a může si zvolit zda chce úlovky řadit dle délky či váhy (od největší po nejmenší).

3.4.1. Záznam úlovku

Po kliknutí tlačítka zaznamenat úlovek, uživatel vyplní ve formuláři pole druh, délku, váhu, vybere typ lovu, vybere lokalitu a popřípadě nahraje obrázky. Uživatel může úlovek také nastavit jako soukromý, stejně jako u lokalit, buď jen pro určitou skupinu, či jen pro sebe.

3.4.2. Detail úlovku

Uživateli se zobrazí informace o daném úlovku, obrázky, a má možnost úlovek komentovat. Má také možnost reagovat na komentáře od jiných uživatelů.



Obrázek 4 - detail úlovku [zdroj: autor]

3.5. Závody

Závody jsou určené pro snazší organizaci rybářských závodů a jejich vyhodnocení. Závod má 4 role, Zakladatel, závodník, měřič a pozorovatel. Na stránce závody se uživateli zobrazí výpis všech veřejných závodů + soukromé závody ve kterých má nějakou roli, tlačítko na vytvoření nového závodu a u každého vypsaného závodu tlačítko detailu.

3.5.1. Nový závod

Pro založení nového závodu stačí ve formuláři vyplnit jen název, vybrat lokalitu, zvolit zda je závod soukromý či ne a vyplnit čas začátku a konce závodu.

3.5.2. Správa závodu

Zakladatel závodu má na stránce detailu závodu 4 tlačítka

- Přidat účastníka: Aby se mohl závod uskutečnit, je potřeba do něj přidat závodníky. To uděláme kliknutím na tlačítko přidat účastníka, vyplníme formulář, stačí Jméno, příjmení a datum narození. Tímto způsobem přidáme všechny závodníky.
- Přidat měřiče: Kliknutím na tlačítko přidat měřiče se nám zobrazí stránka s výpisem měřičů a formulář kde můžeme vybrat uživatele a udělit mu roli měřiče. Měřič narozdíl od závodníka musí být zaregistrovaný uživatel. Měřiči mají důležitou roli ve správě závodů a zapisují závodníkům úlovky
- Přidat pozorovatele: možnost vybrat uživatele kteří budou moci nahlížet na daný závod. Pozorovatel nemá žádné jiné pravomoci než si zobrazit data ohledně závodu a závodníků.
- Ukončit závod: Ukončí závod a znemožní jakýkoliv další zápis, např úlovků.

Měřiči na stránce detailu závodu vidí tlačítko zapsat úlovek pomocí něhož mohou zapisovat úlovky závodníkům. Ve formuláři vyberou závodníka, který chytil rybu, zapíšou její druh, délku, váhu a počet bodů který uživatel dostane.

3.5.3. Detail závodu

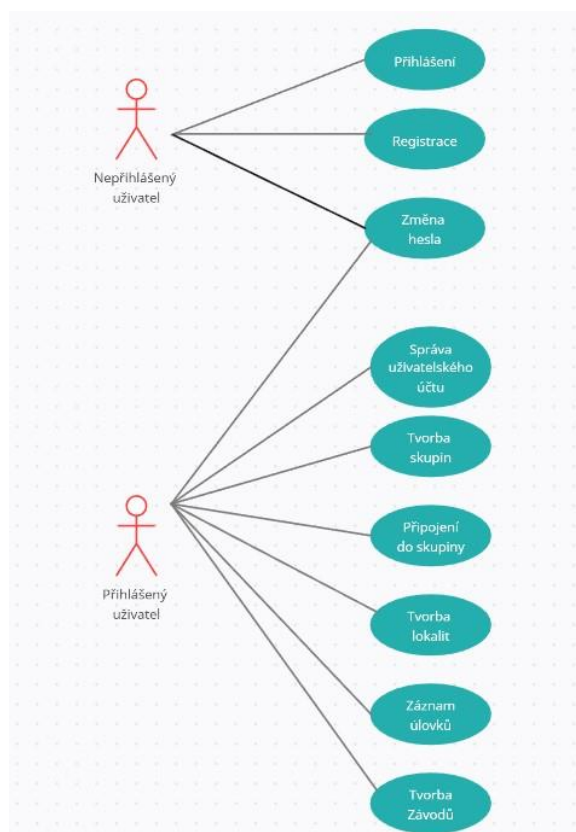
Na stránce detailu závodu normální uživatel/pozorovatel vidí pouze výpis závodníků a jejich aktuální počet bodů. Pro bližší informace o úlovcích závodníka, může kliknout na tlačítko úlovky, a tím se mu zobrazí tabulka se všemi zapsanými rybami které závodník chytil, včetně její délky, váhy, a kolik bodů mu daný úlovek přinesl.

Je zde také vidět “TOP závodu” kde jsou vypsané nejlepší úlovky závodu, podle délky, váhy, a bodů.

4. Use case diagram

Zde jsou zobrazeny UML use case diagramy, pro přehlednost byly rozděleny do tří obrázků.

Use case diagram aplikace



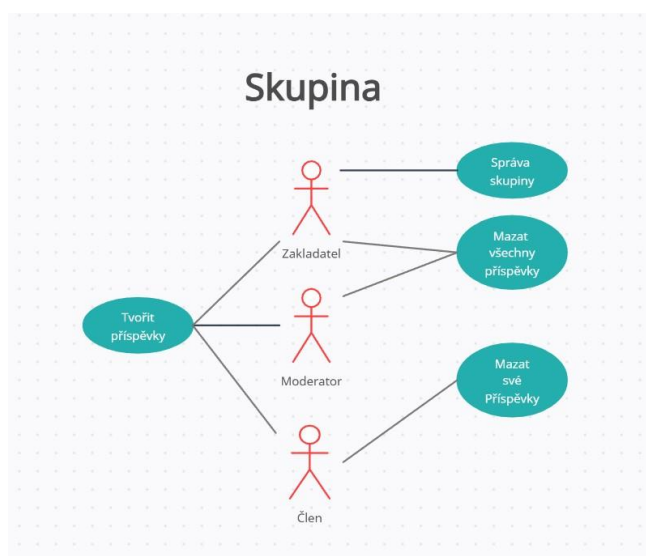
Obrázek 5 Use case diagram aplikace [zdroj: autor]

Na tomto obrázku (Obrázek 5) můžeme vidět use case diagram pro přihlášeného a nepřihlášeného uživatele.

Jak z obrázku vyplívá, jediné akce, které může nepřihlášený uživatel provádět jsou Přihlášení, registrace, či změna hesla pokud-li ho uživatel zapomněl, ale má již registrovaný účet na jeho email. Změna hesla pro nepřihlášeného uživatele funguje tak, že zadá email, na který mu přijde odkaz, pomocí něhož si bude moci změnit heslo.

Přihlášený uživatel může již plně využívat aplikaci, může spravovat svůj účet (změna jména, emailu, hesla, dvoufázové ověření), připojit se již do vytvořené skupiny, či si vytvořit vlastní. Dále může také zaznamenávat své rybářské úlovky, zaznamenávat nové lokality a vytvořit si svůj vlastní rybářský závod.

Use case diagram skupiny



Obrázek 6 - Use case diagram skupiny [zdroj: autor]

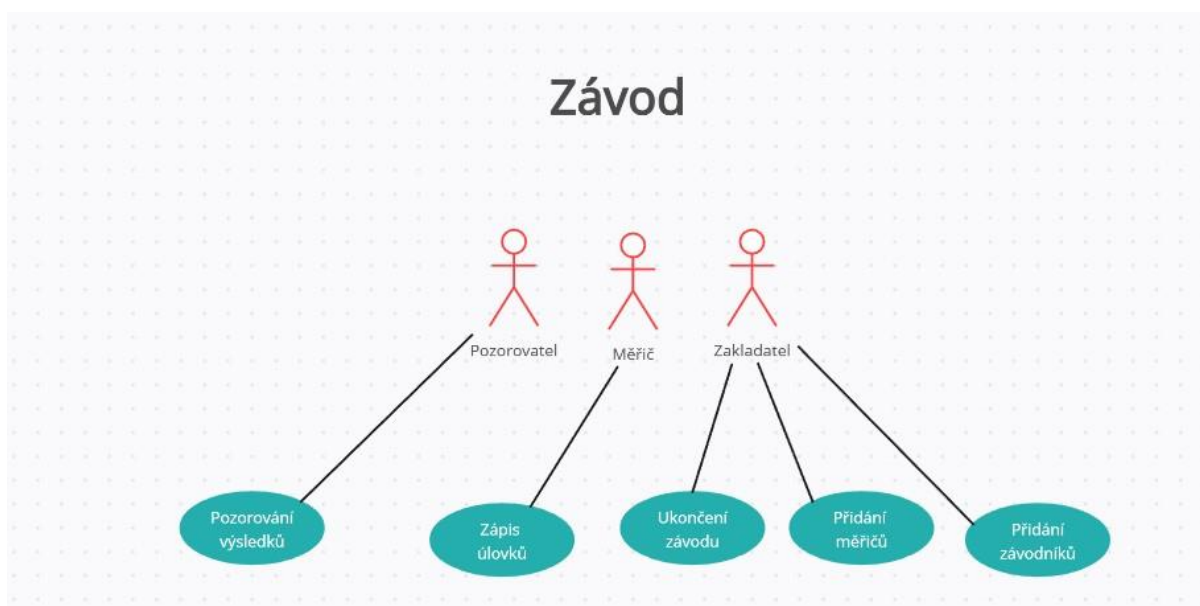
Na tomto obrázku (Obrázek 6), vidíme usecase diagram skupiny.

Skupina má 3 možné role, běžného člena, který může pouze tvořit příspěvky, komentovat je, ale mazat může pouze své vlastní,

Moderátora, ten může mazat jakékoliv příspěvky ve skupině a nakonec

Zakladatele skupiny, jenž může spravovat danou skupinu, čili může generovat pozvánky, udělovat moderátory a mazat všechny příspěvky.

Use case diagram závodu



Obrázek 7 - Use case diagram závodu [zdroj: autor]

Poslední Use case diagram (Obrázek 7) je zaměřený na závody.

Každý závod má 3 role uživatelů, Zakladatel může na začátku přidat závodníky (závodníci vůbec nemusí mít v aplikaci účet, ani o ní vědět, stačí když je zakladatel zapíše do databáze), přidá měřiče a pozorovatele (měřiči i pozorovatelé musí mít účet).

Měřiči mají jedinou speciální pravomoc, tou je zápis úlovků závodníkům, když závodník chytí rybu, on k jeho jménu připíše úlovek s detaily (název, body, délka, váha).

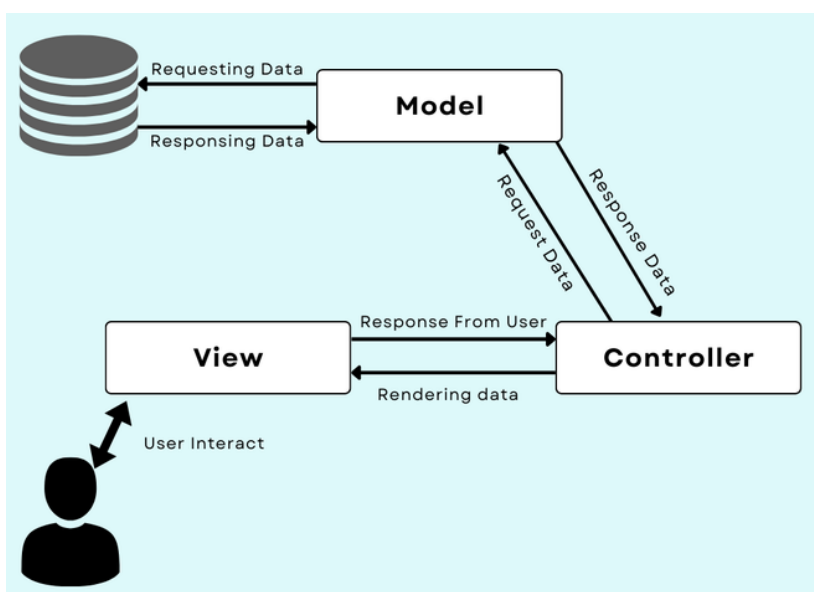
Pozorovatel má jedinou pravomoc, jak již z názvu vyplívá, může pouze pozorovat výsledky závodu (např když je závod soukromý a normální uživatel aplikace k nim nemá přístup).

5. Architektura

Aplikace je postavena na frameworku Laravel. Autentizaci provádím pomocí JetStreamu. Projekt je založen na architektuře MVC, kde Controllery zpracovávají požadavky a předávají data Blade šablonám pro zobrazení. Pro práci s databází je využit Eloquent ORM, přičemž jako databázový systém je použita SQLite. Pro styling aplikace je využit Tailwind CSS a flowbite komponenty. Jedním z klíčových prvků Laravelu je také systém routování, který umožňuje efektivně směřovat HTTP požadavky na příslušné části aplikace. (Laravel, 2025)

Aplikace využívá architekturu Model-View-Controller (MVC), která zajišťuje přehledné oddělení logiky, dat a uživatelského rozhraní.

- **Model (M):** Modely představují datovou vrstvu aplikace a pracují s databází prostřednictvím Eloquent ORM. Každý model odpovídá konkrétní tabulce v databázi a umožňuje snadné provádění operací, jako je čtení, zápis, aktualizace a mazání záznamů. Například model *Zavod* reprezentuje závody a obsahuje metody pro práci s nimi.
- **Controller (C):** Kontrolery zpracovávají požadavky od uživatelů, a probíhá v nich „logika“ aplikace. Volají odpovídající modely a vracejí odpovědi ve formě Blade šablon. Například *ZavodyController* načítá seznam závodů z databáze a předává je do šablony, kde jsou zobrazeny.
- **View (V):** Pohledy (Views) jsou zodpovědné za prezentaci dat uživateli. V aplikaci jsou implementovány pomocí Blade šablon, které umožňují využití dynamického obsahu. Blade také podporuje komponenty a layouty, což usnadňuje správu vzhledu aplikace.



Obrázek 8 - MVC architektura [zdroj: xxx (<https://nitsantech.com/blog/laravel-development-the-complete-guide/>)]

5.1. Základní koncept routování

V Laravelu jsou routy definovány v souboru `routes/web.php`. Každá ruta specifikuje HTTP metodu (GET, POST, atd.), URI cestu a akci, která se má provést při přístupu na danou cestu. (Laravel, 2025) (Laravelblog.cz, 2025)

Příklad routy z projektu

```
Route::get('/skupiny/{id}', [SkupinaController::class, 'show'])->name('skupiny.show');
```

Obrázek 9 - Příklad routy [zdroj: autor]

Tato ruta:

- Reaguje na GET požadavky
- Očekává URL ve formátu `/skupiny/{id}`, kde `{id}` je dynamický parametr
- Směřuje požadavek na metodu `show` v `SkupinaController`
- Má přiřazený název `'skupiny.show'` pro snadné generování URL

Jak routy fungují

1. **Zpracování požadavku:** Když uživatel přistoupí na URL `/skupiny/5`, Laravel zachytí tento požadavek.
2. **Nalezení odpovídající routy:** Framework prohledá definované routy a najde shodu s `/skupiny/{id}`.
3. **Extrakce parametrů:** Hodnota `'5'` je extrahována jako parametr `id`.
4. **Volání controlleru:** Laravel zavolá metodu `show` v `SkupinaController` a předá jí hodnotu `id`.
5. **Zpracování a odpověď:** Controller zpracuje požadavek (načte data skupiny s ID 5) a vrátí odpověď (view s daty).

```

public function store(Request $request)
{
    $validated = $request->validate([
        'nazev' => 'required|string',
        'lokalita' => 'nullable|exists:lokalita,id',
        'soukromost' => 'boolean',
        'datum_zahajeni' => 'required|date',
        'datum_ukonceni' => 'required|date|after_or_equal:datum_zahajeni',
    ]);

    $zavod = Zavod::create([
        'nazev' => $validated['nazev'],
        'id_zakladatele' => auth()->id(),
        'lokalita' => $validated['lokalita'] ?: null,
        'soukromost' => $request->has( key: 'soukromost'),
        'stav' => 1,
        'datum_zahajeni' => $validated['datum_zahajeni'],
        'datum_ukonceni' => $validated['datum_ukonceni'],
    ]);

    return redirect()->route( route: 'zavody.index')->with('success', 'Závod byl úspěšně přidán.');
```

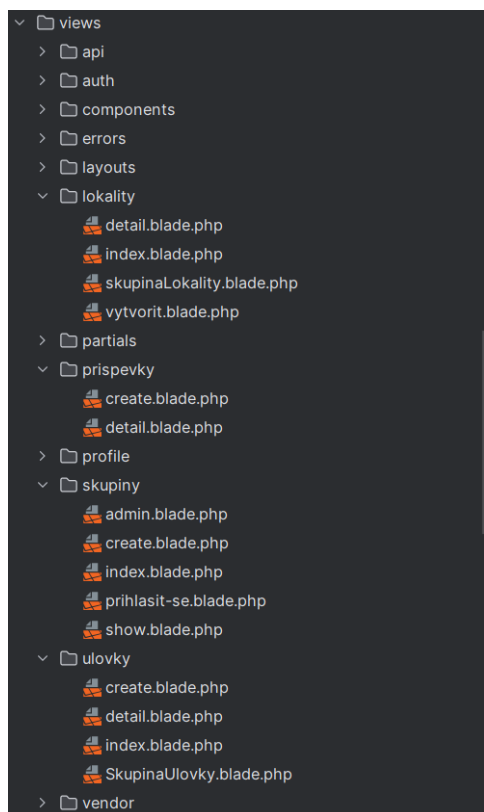
Obrázek 10 - metody pro vytvoření nového závodu [zdroj: autor]

Příklad metody pro vytvoření nového závodu, soubor ZavodyController.

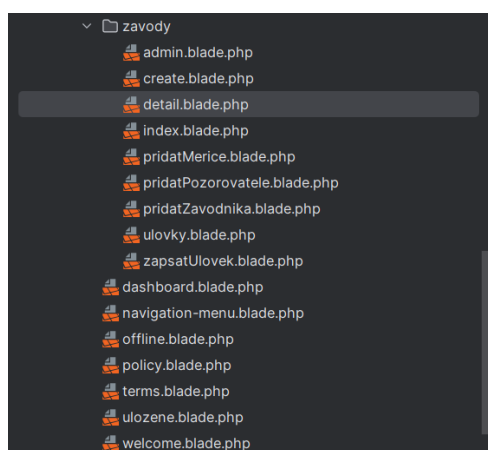
5.2. Struktura projektu

V této části jsou obrázky, zobrazující strukturu mého projektu.

Views



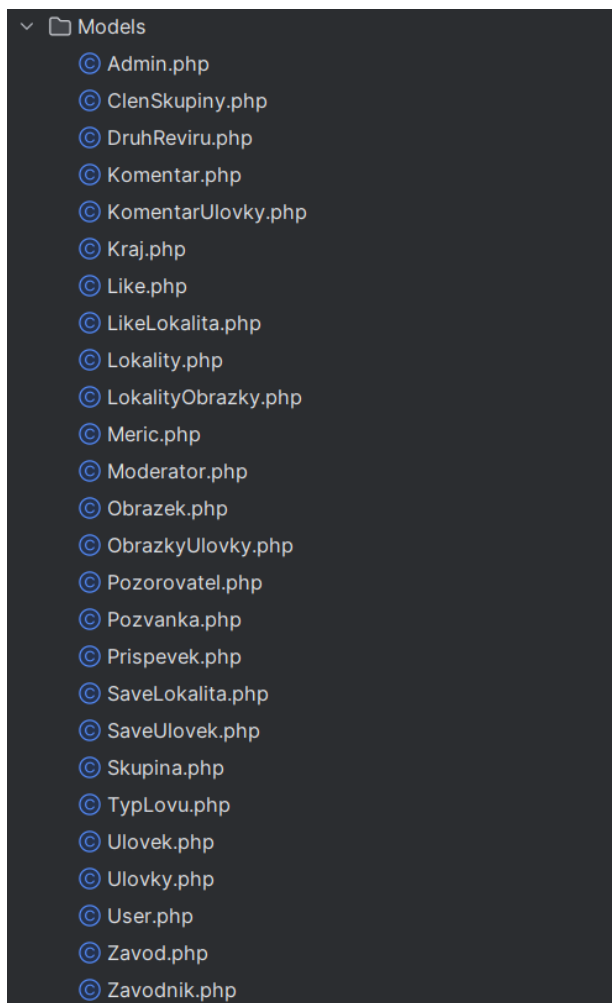
Obrázek 11 - views soubory 1. část [zdroj: autor]



Obrázek 12 - views soubory 2. část [zdroj: autor]

Na výše zobrazených obrázcích (Obrázek 11 a Obrázek 12) můžeme vidět strukturu views souborů, jenž jsou zodpovědné za prezentaci dat uživateli. Soubory jsou rozdělené do složek a podsložek pro lepší přehlednost.

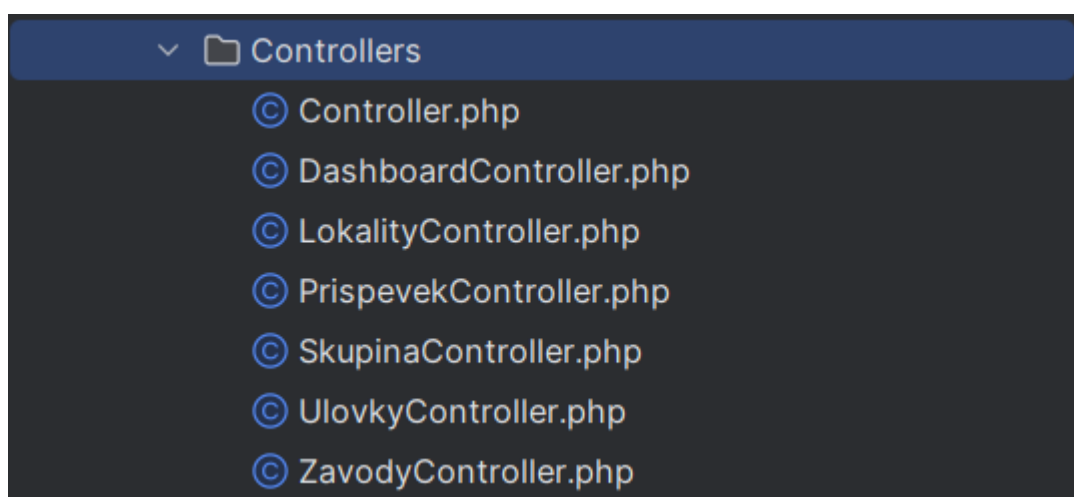
Models



Obrázek 13 - Models soubory [zdroj: autor]

Obrázek 13 nám zobrazuje modely aplikace. Každý model má také svoji tabulku v databázi. Jsou určeny pro snazší, rychlejší a bezpečnější práci s daty z databáze.

Controllers



Obrázek 14 - Controllers soubory [zdroj: autor]

Controllery, pro můj projekt jedna z nejdůležitějších částí, v nich se odehrává většina logiky. Kvůli přehlednosti mám metody rozděleny do několika souborů. Např. v souboru `ZavodyController.php` jsou jen metody které se starají o závody.

5.3. Uložiště

Ve svém projektu používám jako databázi SQLite, což je kompaktní, serverless relační databázový systém. Jeho hlavní výhodou je jednoduchost, protože nevyžaduje žádnou konfiguraci ani správu serveru. Celá databáze je uložena v jediném souboru, což usnadňuje zálohování a přenos dat. SQLite je ACID kompatibilní, což zajišťuje spolehlivost a integritu dat. Také má nízké nároky na zdroje, což ji činí ideální pro vestavěné systémy a mobilní aplikace.

Má však i svá omezení – podporuje pouze jednoho zapisujícího v daném okamžiku, což může být problém při vyšší zátěži. Není tedy vhodná pro velmi velké databáze nebo projekty s vysokým počtem současných uživatelů. Některé pokročilé funkce běžné u plnohodnotných RDBMS zde také chybí. Pro můj projekt však SQLite stačí, ale pokud by aplikace měla v budoucnu mnoho uživatelů, bylo by vhodné přejít na jinou databázi. (Wikipedia-Sqlite, 2025) (Sqlite, 2025)

5.4. Struktura databáze

V příloze číslo 1 můžeme vidět strukturu mé databáze, a vztahy mezi jednotlivými tabulkami. Jako příklad vyberu pár tabulek a popíšu vztahy mezi nimi. Pro demonstraci zvolím tabulky ohledně funkce Závodů.

Tabulka **Závody**

Obsahuje základní údaje k vytvoření závodu

Tabulka závody má pole:

ID (unikátní identifikátor pro rozlišení mezi závody)

Název (string – Název souboru)

Id_zakladatele (cizí klíč na tabulku users, do tohoto pole se zapíše ID uživatele z tabulky users co založil tento závod)

Lokalita (cizí klíč na tabulku lokality, určuje místo, kde se závod koná)

Soukromost (Bool, určuje, zda je závod soukromý či ne)

Stav (cizí klíč na tabulku stavy_zavodu, určuje stav závodu)

Datum zahájení (timeStamp, čas a datum kdy závod začíná)

Datum ukončení (timeStamp, čas a datum kdy závod končí)

Tabulka **cleni_zavodu**

Do ní se zapisují závodníci a jejich údaje. Podle Id_zavodu se následně spojí s daným závodem.

Obsahuje pole:

ID (unikátní id pro rozlišení mezi závodníky)

Id_zavodu (cizí klíč, pro spojení závodníka se závodem)

Jmeno (string, jméno závodníka)

Prijmeni (string, příjmení závodníka)

Datum narození (date, datum narození závodníka, je zde k rozlišení mezi závodníky, např se stejnými jmény)

Tabulka **ulovky_zavodu**

Do této tabulky se zapisují úlovky, zapisují je měřiči. Každý úlovek je přiřazen k závodu, k uživateli a k měřiči.

Obsahuje pole:

ID

ID_zavodu (cizí klíč na tabulku zavody, je zde k tomu aby se určilo do jakého závodu tento úlovek patří)

ID_zavodnika (cizí klíč na tabulku clení_zavodu, slouží k určení toho, jaký rybář úlovek ulovil)

ID_merice (cizí klíč na tabulku merici, slouží k určení jaký měřič úlovek zapsal)

Druh_ryby (string, druh ryby která byla ulovena)

Delka (integer, délka ryby)

Vaha (integer, váha ryby)

Body (integer, zapisuje se zde jakou hodnotu má daný úlovek)

Indexace

V tabulce používám indexaci, ta slouží k rychlejšímu nalezení výsledků.

Když je v databázi mnoho záznamů, např. v tabulce úlovky mnoho úlovků, a já hledám jen úlovky od konkrétního rybáře, tak neprohledávám celou tabulku, jen skočím rovnou na místo, kde jsou dané záznamy.

Tabulka merici

Tabulka merici je určená pro zapsání a přidělení měřičů k závodu. Je velice jednoduchá takže má jen pár polí.

ID

ID_zavodu (cizí klíč na tabulku zavody, slouží k tomu, aby se měřič přiřadil k danému závodu)

ID_uzivatele (cizí klíč na tabulku users, slouží k určení uživatele)

5.5. Ukládání souborů

Laravel poskytuje robustní a flexibilní systém pro práci se soubory, včetně jejich ukládání. V uvedeném příkladu je demonstrováno, jak Laravel zpracovává a ukládá obrázky.

Ukládání souborů

Každý soubor je uložen pomocí metody `store()`:

```
$cesta_k_obrazku = $obrazek->store('obrazky', 'public')
```

Obrázek 15 - příklad ukládání obrázků [zdroh: autor]

- První argument ('obrazky') specifikuje adresář, do kterého se soubor uloží.
- Druhý argument ('public') určuje disk pro ukládání, v tomto případě veřejný disk.

Generování unikátních názvů

Laravel automaticky generuje unikátní názvy pro ukládané soubory, aby se zabránilo přepsání existujících souborů

Ukládání cesty do databáze

Po uložení souboru je cesta k němu uložena do databáze:

```
Obrazek::create([
    'ID_prispevku' => $prispevek->id,
    'cesta_k_obrazku' => $cesta_k_obrazku,
])
```

Obrázek 16 - příklad ukládání obrázků [zdroj: autor]

Výhody tohoto přístupu

1. **Bezpečnost:** Automatické generování názvů souborů zabraňuje konfliktům a potenciálním bezpečnostním rizikům.
2. **Flexibilita:** Laravel umožňuje snadno konfigurovat různé disky pro ukládání (lokální, Amazon S3, atd.).
3. **Snadná integrace:** Ukládání souborů je úzce integrováno s Eloquent ORM, což usnadňuje práci s databází.
4. **Efektivita:** Laravel optimalizuje proces ukládání pro rychlost a efektivitu.

Závěr

Vytvořená aplikace splňuje požadavky zadání — nabízí správu veřejných a soukromých skupin s přístupem chráněným heslem, možnost sdílení úlovků a lokalit, a vytváření komunikačních vláken. Navíc jsem na základě zpětné vazby od uživatelů rozšířil projekt o funkcionalitu pro organizaci rybářských závodů, což původně nebylo součástí zadání. Tato rozšířená část zahrnuje možnost rychlého záznamu úlovků během závodů, přehled závodníků a průběžné výsledky v reálném čase. Taková funkce na trhu chyběla, a proto jsem se rozhodl ji do aplikace integrovat.

Přestože aplikace splňuje svůj základní účel, stále je prostor pro zlepšení a pro přidání dalších funkcionalit.

Do budoucna by se dal projekt více rozvíjet několika směry, např. pro větší užitek místním rybářským organizacím, např formou organizací brigád či objednání povolenek.

Celkově si myslím že projekt nabízí efektivní a moderní řešení pro rybářské skupiny či místní organizace, které hledají způsob, jak zjednodušit organizaci závodů a propojit se s ostatními rybáři.

K posouzení předkládám hotovou webovou aplikaci společně s dokumentací

Zdroje

Adermann, Nils a Boggiano, Jordi. 2025. Composer. [Online] 2025. [Citace: 3. 25 2025.] <https://getcomposer.org/doc/00-intro.md>.

European Carp Fishing Association. 2025. [fishingstatistics.com](https://www.fishingstatistics.com/). [Online] 2025. [Citace: 25. 3 2025.] <https://www.fishingstatistics.com/>.

FishAngler. 2025. FishAngler, LLC. [Online] 2025. [Citace: 25. 3 2025.] <https://www.fishangler.com/>.

Fishsurfing. 2025. Fishsurfing. *Fishsurfing*. [Online] 2025. [Citace: 25. 3 2025.] <https://www.fishsurfing.com/cs/>.

Jalli, Artturi. 2022. Built In. *Built In*. [Online] 2022. [Citace: 25. 3 2025.] <https://builtin.com/software-engineering-perspectives/laravel>.

JetStream. 2025. JetStream. [Online] 2025. [Citace: 24. 3 2025.] <https://jetstream.laravel.com/introduction.html>.

Laravel. 2025. Laravel. [Online] 2025. [Citace: 25. 3 2025.] <https://laravel.com/>.

—. **2025.** Laravel routing. [Online] 2025. [Citace: 25. 3 2025.] <https://laravel.com/docs/12.x/routing>.

Laravelblog.cz. 2025. laravelblog.cz. [Online] 2025. [Citace: 23. 3 2025.] <https://laravelblog.cz/dokumentace/6.x/routing>.

Michálek, Martin. 2021. <https://www.vzhurudolu.cz/>. [Online] 2021. [Citace: 25. 3 2025.] <https://www.vzhurudolu.cz/prirucka/tailwind-css>.

Souček, Vladimír. 2024. Google play - Vranka.cz. [Online] 2024. [Citace: 25. 3 2025.] <https://play.google.com/store/apps/details?id=cz.vranka.rybarivranka>.

Sqlite. 2025. <https://www.sqlite.org/>. [Online] 2025. [Citace: 25. 3 2025.] <https://www.sqlite.org/>.

Vymětal, Vojtěch. 2024. Rybářské návody. [Online] 20. 11 2024. [Citace: 25. 3 2025.] <https://www.rybarske-navody.cz/2024/11/20/vranka-cz-konci-oblibena-aplikace-pro-rybare-ukoncila-provoz/>.

w3schools. 2025. [w3schools.com](https://www.w3schools.com/). [Online] 2025. [Citace: 25. 3 2025.] https://www.w3schools.com/whatis/whatis_npm.asp.

Wikipedia-Sqlite. 2025. *Wikipedia*. [Online] 2025. [Citace: 25. 3 2025.] <https://en.wikipedia.org/wiki/SQLite>.

Seznam obrázků

Obrázek 1 - Dashboard stránka [zdroj: autor]	12
Obrázek 2 - Stránka skupiny [Zdroj: autor]	13
Obrázek 3 Admin panel skupiny [zdroj: autor]	14
Obrázek 4 - detail úlovku [zdroj: autor]	16
Obrázek 5 Use case diagram aplikace [zdroj: autor]	18
Obrázek 6 - Use case diagram skupiny [zdroj: autor]	19
Obrázek 7 - Use case diagram závodu [zdroj: autor]	20
Obrázek 8 - MVC architektura [zdroj: xxx (https://nitsantech.com/blog/laravel-development-the-complete-guide)]	21
Obrázek 9 - Příklad routy [zdroj: autor]	22
Obrázek 10 - metody pro vytvoření nového závodu [zdroj: autor]	23
Obrázek 11 - views soubory 1. část [zdroj: autor]	24
Obrázek 12 - views soubory 2. část [zdroj: autor]	24
Obrázek 13 - Models soubory [zdroj: autor]	25
Obrázek 14 - Controllers soubory [zdroj: autor]	26
Obrázek 15 - příklad ukládání obrázků [zdroh: autor]	29
Obrázek 16 - příklad ukládání obrázků [zdroj: autor]	29

