

Zásilkárna

Autoři: Jakub Smejkal(xsmejk28), Klára Formánková(xforma14)

Odkaz na aplikaci, spuštěnou na domácím serveru:

<http://smarthomeapp.duckdns.org:8000>

Implementace zadání [Tým 09] Úložiště souborů (Ing. Bednář/Ing. Koutenský).

Projekt jsme implementovali po vzoru služby www.uschovna.cz, která se zadání podobá, prozkoumali jsme jak funguje a následně jsme implementovali vlastní řešení. Službu jsme implementovali tak aby podporovala pouze nahrávání jednoho souboru, podle zadání. S pomocí použitých nástrojů by bylo možné toto řešení rozšířit i pro více souborů.

Pro popis instalace a spuštění projektu viz. README.md

Rozdělení práce v týmu

xforma14

- Průzkum a získávání informací pro projekt
- Návrh projektu
- Design a implementace frontendu
- Návrh databázové tabulky
- Dokumentace frontendu

xsmejk28

- Průzkum a získávání informací pro projekt
- Návrh projektu
- Implementace backendu a spojení s externími knihovnami
- Dokumentace backendu

Použité technologie a knihovny

Laravel

Pro vytvoření projektu jsme využili php framework Laravel. Tento framework využívá MVC architekturu pro implementování webových stránek a aplikací. Modelem se rozumí většinou tabulka v databázi. Controller je php skript starající se o vytváření databázových záznamů a další funkce backendu a View zahrnuje *.blade.html soubory, ve kterých je mimo běžného HTML možné použít i speciální konstrukce jako například podmínky nebo cykly.

Tento framework dále umožňuje využít i rozšiřující balíčky, jako například v našem projektu použitý Laravel Chunk Upload.

Laravel byl zvolen z důvodu předchozích zkušeností při implementaci jiných projektů. Dalšími faktory byly externí balíčky a možnost praktičtějšího zápisu HTML souborů.

Posledním faktorem byla možnost rychlého nasazení do služby Heroku, která umožňuje aktualizaci stránek z GitHub, čímž nám usnadnila práci v týmu.

Laravel Chunk Upload¹

Toto rozšíření pro framework Laravel umožňuje spojení knihovny resumable.js využívané ve view a php kódu v Controlleru. Tento balíček není nutný v případě omezení velikosti nahrávaného souboru.

Resumable.js²

Knihovna umožňující rozdělení nahrávaného souboru na menší části při nahrávání přes HTTP. Implementuje bezpečné nahrávání, které je možné pozastavit/obnovit. Při chybě nahrávání menších podčástí souborů je toto nahrávání zopakováno, tudíž knihovna počítá i s menšími výpadky.

Jedná se o jednu z možných knihoven, které lze spojit s Laravel Chunk Upload rozšířením.

Bootstrap

Pro efektivní tvorbu frontendu aplikace je využit jeden z nejpoužívanějších nástrojů pro stylování - knihovna Bootstrap. Tato knihovna poskytuje sadu předpřipravených, již nastýlovaných komponent uživatelského rozhraní. Použití takových komponent značně urychlilo vývoj frontendu aplikace.

Backend

Pro nahrávání souborů jsme po úvodním testování bez externích knihoven, které omezovalo velikost nahrávání větších souborů, využili balíček Laravel Chunk Upload s knihovnou resumable.js. Tento problém by bylo možné vyřešit i pomocí úpravy hodnot, které omezují maximální velikost nahrávaných souborů v php.ini, toto řešení ovšem může zpomalit aplikaci. Pomocí použitého řešení je větší soubor rozdělen na menší části (chunks), tyto části jsou nahrány a po konci nahrávání jsou opět spojeny do původního souboru. Pro spojení Laravel Chunk Upload byly využity příklady přímo z GitHub repozitáře Laravel Chunk Upload¹ a z webového portálu StarTutorial³, které byly upravené pro potřeby našeho projektu.

¹ Martin Kluska 2016, *pionl/laravel-chunk-upload*, GitHub, navštíveno 14.3.2022, <https://github.com/pionl/laravel-chunk-upload>

² <https://github.com/23/resumable.js>

³ *Resumable file upload*, StarTutorial, navštíveno 14.3.2022, <https://startutorial.com/view/resumable-file-upload-part-1>

Soubory jsou ukládány do složky “/storage/files” pod náhodně vygenerovaným názvem o délce 30 znaků s původní příponou. Tyto soubory jsou uloženy po dobu 30 dnů, ale dobu je možné prodloužit v prostředí správce.

Spolu s nahráním souboru do úložiště je vytvořen záznam v databázi, kde jsou uvedeny všechny informace potřebné k znovuzískání souboru v původní podobě. Dále jsou v databázi uloženy odkazy pro uživatele a pro správce spolu s datem expirace a počtem stažení.

File	
PK	<u>file_link VARCHAR NOT NULL</u>
	admin_link VARCHAR NOT NULL file_storage_path VARCHAR NOT NULL original_name VARCHAR NOT NULL extension VARCHAR NOT NULL number_of_downloads INT delete_date DATE NOT NULL

Návrh tabulky pro uložený soubor

Pro mazání souborů s vypršenou dobou platnosti je využito Laravel scheduleru, který umožňuje definovat, které funkce se mají spouštět v jakém intervalu. Scheduler je na serveru spuštěn pomocí Cron Job. Při spuštění příkazu “php artisan schedule:run” jsou zkontrolovány všechny funkce definované v scheduleru, a pokud vypršela definovaná doba od posledního spuštění, je funkce spuštěna. Funkce mazání souborů je spuštěna každý den.

Frontend

Při implementaci frontendu aplikace bylo cílem vytvořit velmi jednoduché uživatelské prostředí, které nabídne uživateli rychlý přístup ke všem operacím, které může provést. Veškeré komponenty uživatelského rozhraní jsme se proto rozhodli umístit do tzv. karty - vyznačeného obdélníku, který vypadá na všech pohledech aplikace stejně a nachází se na stejném místě obrazovky. Mění se pouze obsah a výška karty. Uvnitř karty se vždy nachází pouze nezbytné prvky, které může uživatel potřebovat.

Pro efektivní stylování byly využity komponenty knihovny Bootstrap, jako například tlačítka nebo ikony. Rozmístění komponent v kartě a styl některých vlastních komponent byl definován pomocí čtyř CSS souborů. Styly společné napříč aplikací jsou definovány v souboru style.css. Další tři CSS soubory (home.css, file_view.css a admin_view.css) definují styly jednotlivých pohledů. Funkčnost některých komponent (např. tlačítka pro zkopírování odkazu) je implementována v jazyce JavaScript. Soubor dropArea.js uchovává kód pro funkci drag and drop komponenty a soubor copyToClipboard.js obsahuje implementaci chování tlačítek pro kopírování odkazů.

Po vzoru dnes už většiny webových aplikací jsme se rozhodli mimo tradiční výběr souboru ke sdílení přes průzkumníka souborů umožnit uživateli také nahrát soubor pomocí tzv. drag and drop způsobu. Abychom nevymýšleli již vymyšlené, inspirovali jsme se jedním z existujících řešení⁴, které jsme upravili tak, aby bylo vhodné pro naše potřeby a komunikovalo se zbytkem našeho kódu.

Plocha, kam může být soubor “přetažen a upuštěn”, je označena jako drop-area a jsou nad ní definovány specifické události, např. dragenter, dragover, dragleave, drop. Tyto události umožňují měnit vzhled a chování drop-area. Sled událostí a následných reakcí na ně je vzhledem k očekávanému chování přímočarý:

- pokud je soubor tažen nebo držen nad drop-area, plocha je zvýrazněna,
- pokud soubor opustí drop-area, plocha přestane být zvýrazněna,
- pokud je soubor upuštěn nad drop-area, proběhne nahrání souboru a plocha přestane být zvýrazněna.

Pro jednoduché sdílení odkazů bylo ke každému odkazu přidáno tlačítko pro zkopírování odkazu. Po kliknutí na tlačítko je zavolána metoda `navigator.clipboard.writeText()`, která zkopíruje odkaz do schránky, stejně jako při označení celého odkazu a zkopírování přes klávesovou zkratku `CTRL+C`.

Výsledná aplikace

Prvním krokem uživatele, který chce sdílet soubor přes Zásilkárnu, je nahrání souboru. Uživatel může zvolit variantu klasického výběru souboru z průzkumníka souborů, který se otevře po kliknutí na tlačítko ‘Vybrat soubor’ na úvodní stránce aplikace, nebo může jednoduše přetáhnout soubory do vyznačeného obdélníku.




Po zvolení souboru se objeví tlačítko pro odeslání, které provede nahrání souboru (chunk upload) do úložiště serveru. Při nahrávání na server je uživateli zobrazen progress bar, který ukazuje, že aplikace soubor opravdu nahrává a v jakém stavu nahrávání je.

Uživatel je po nahrání souboru přesměrován do prostředí správce. Toto prostředí umožňuje všechny funkce prostředí běžného uživatele spolu s možností změnit datum expirace souboru, smazání souboru či generování nových odkazů pro uživatele/správce. Pro jednodušší distribuci odkazu bylo přidáno tlačítko pro rychlé kopírování odkazů.

⁴ Joseph Zimmerman 2018, *How To Make A Drag-and-Drop File Uploader With Vanilla JavaScript*, Smashing Magazine, navštíveno 8.4.2022, <https://www.smashingmagazine.com/2018/01/drag-drop-file-uploader-vanilla-js/>

Zásilkárna


Zásilka úspěšně odeslána!

 flowers.jpg  

Staženo: 0x


Uloženo do: 20.05.2022 Uložit

Odkaz pro sdílení:

<http://127.0.0.1:8000/uGbrswV7kwm1dmOI4YsFo1qpxwzLfl> 

Nový odkaz

Administrátorský odkaz:



<http://127.0.0.1:8000/uGbrswV7kwm1dmOI4YsFo1qpxwzLfl/tQoi5zllLyINXcS> 

Nový admin odkaz

Běžný uživatel po navštívení odkazu pro sdílení vidí název souboru s příponou, počet stažení tohoto souboru a má možnost si tento soubor stáhnout.

Zásilkárna

Vaše zásilka

 flowers.jpg 

Staženo: 0x