

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Webová aplikace pro tvorbu prezentací



2018

Vedoucí práce: Mgr. Martin Tr-
nečka, Ph.D.

Petr Valigura

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor:	Petr Valigura
Název práce:	Webová aplikace pro tvorbu prezentací
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2018
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	Mgr. Martin Trnečka, Ph.D.
Počet stran:	37
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliographic info

Author:	Petr Valigura
Title:	Web application for slideshows creation
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2018
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Martin Trnečka, Ph.D.
Page count:	37
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Práce pojednává o vývoji webové aplikace pro vytváření prezentací. Tato aplikace, která je její součástí, je počítačový program typu klient-server. Klient, implementován pomocí knihovny React, nabídne uživateli možnost zdarma vytvářet, upravovat a prezentovat slajdy s multimediálním obsahem. Server, psaný v Node.js, pak umožní tyto slajdy perzistentně ukládat. Práce také obsahuje zhodnocení podobných aplikací, popis implementačních detailů a manuál pro běžného uživatele.

Synopsis

Thesis deals about development of a web application for slideshows creation. The application, which is part of this thesis, is a client-server computer program. The client that is implemented in React offers options for creating, editing and presenting slides with multimedia content. The server which is written in Node.js offers persistent storage for these slides. The thesis contains reviews of similar applications, a description of implementation and a user manual.

Klíčová slova: webová aplikace; prezentace; JavaScript; MongoDB

Keywords: web application; presentation; JavaScript; MongoDB

Děkuji vedoucímu mé práce Mgr. Martinu Trnečkovi, Ph.D. za jeho ochotu, nápady a rady, které mi při tvorbě aplikace pomohly. Také moc děkuji své rodině za podporu nejen při psání práce, ale i při studiu.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
1.1	O aplikaci	8
1.2	Motivace	8
2	Existující aplikace	10
2.1	PowerPoint	10
2.2	Prezi	11
2.3	Prezentace Google	12
2.4	Srovnání	12
3	Použité technologie	13
3.1	JavaScript	13
3.2	HTML 5	13
3.3	CSS 3	14
3.4	Node.js	14
3.5	Express	15
3.6	React	15
3.7	Draft.js	16
3.8	MongoDB	16
3.9	Mongoose	17
3.10	html2canvas, jspdf	18
3.11	Web Font Loader	18
3.12	Font Aswesome	18
4	Programátorské dokumentace	19
4.1	Adresářová struktura	19
4.2	Zabezpečení	19
4.3	Databáze	20
4.4	REST API	21
4.5	Služba pro převod obrázků	21
4.6	PdfConverter	22
4.7	Vlastní komponenty	22
5	Uživatelská dokumentace	23
5.1	Rozložení aplikace	23
5.2	Vytvoření slajdu	24
5.3	Změna pořadí slajdů	24
5.4	Zpět, vpřed	24
5.5	Vytvoření prvku	24
5.6	Prezentování	26
5.7	Správa fontů	26
5.8	Pokročilé možnosti	27
5.9	Uložení	28

5.10 Přihlášení	29
5.11 Export, import	29
5.12 PDF	30
6 Rozšíření aplikace	31
Závěr	32
Conclusions	33
A Instalace a spuštění serveru	34
A.1 Požadavky:	34
A.2 Spuštění serveru:	34
B Instalace a spuštění aplikace:	34
B.1 Požadavky	34
B.2 Spuštění aplikace:	34
C Testování	35
D Obsah přiloženého CD/DVD	35
Literatura	36

Seznam obrázků

1	Nabídka funkcí karty Domů.	10
2	Nově vytvořená prezentace v Prezi.	11
3	Rozložení úvodní stránky.	23
4	Příklad textového boxu.	25
5	Příklad okna správce obrázků pro uloženou prezentaci.	26
6	Příklad okna správce fontů.	27
7	Příklad použití konzole.	28
8	Okno uložení již uložené prezentace.	29
9	Pokus o uložení prezentace se špatným heslem.	29
10	Příklad řetězců prezentace v PDF.	30

Seznam zdrojových kódů

1	Definice komponenty Image.	16
2	Definice schématu prezentace.	17
3	Uložený záznam, který neobsahuje slajdy.	20
4	Odpověď při úspěšném uložení prezentace.	21
5	Objekt, podle kterého se vygeneruje okno uložení.	22

1 Úvod

V dnešní době se studující člověk prezentacím jen stěží vyhne. Sám jsem během studia základní školy na vlastní kůži pocítil přechod z prezentování pomocí meotaru na promítání textu pomocí dataprojektoru, kde text po chvíli vystřídaly přehledně, graficky upravené slajdy. Tento systém pak vydržel nejen celou střední, ale také i vysokou školu. Použití prezentací plně využívají jak učitelé, tak i studenti. Využití dokonce nalezne i mimo vzdělávání. Při vytváření prezentací je však nutné použít nějaký software a ten jsem se rozhodl vytvořit. V této části nastíním popis aplikace a motivaci, kterou jsem měl k jejímu vytvoření i přesto, že na trhu je již celá řada existujících řešení.

1.1 O aplikaci

Cílem mé bakalářské práce je vytvořit webovou aplikaci pro tvorbu prezentací, která by si se svou jednoduchostí a přenositelností našla místo ve světě prezentování.

Díky možnosti pracovat s celou aplikací v internetovém prohlížeči a vygenerování slajdů do PDF¹ souboru má uživatel jistotu, že spustí prezentaci téměř všude. Navíc není potřeba se u aplikace registrovat ani přihlašovat. Uživatel je tak zbaven povinnosti zadávat e-mail a jiné osobní údaje. Vždy se ukládá pouze samotná prezentace, která je chráněna heslem. Uložená je po dobu třiceti dní od poslední úpravy v databázi ve formě podobné JSON² zápisu. Pro třicetidenní limit jsem se rozhodl proto, že prezentace, zvláště u studentů, jsou používány jednorázově. Pak, v případě že jsou uloženy v nějaké databázi, jen zbytečně zabírají místo. Ti uživatelé, kteří si prezentaci chtějí uchovat, mohou využít možnosti vygenerování do PDF a nebo exportování do textového formátu, ten si pak mohou uložit na disk a v případě potřeby do aplikace znova importovat.

Název aplikace byl vytvořen hrou s anglickými slovy „slide“ (označení výrazu pro snímek prezentace, v češtině se častěji používá výraz slajd) a „simply“ (jednoduše), jejichž kombinací vyšlo slovo „slidly“, které jsem zvolil jako jméno. I přes tento cize znějící název je aplikace cílena pro české a slovenské uživatele, tudíž uživatel nemusí anglický jazyk umět.

1.2 Motivace

Ve druhém ročníku střední školy, jsme se v hodinách informatiky seznámili i s aplikací PowerPoint (více v podsekcí 2.1), kterou jsme probrali docela podrobně. Byl jsem mile překvapen tím, kolik toho nabízí. Jelikož se jedná o placenou aplikaci, jenž nebyla volně ke stažení, setkával jsem se s ní do té doby pouze jako divák, když jsem sledoval prezentace, které pro nás tvořili učitelé. Ti ale nevyužívali ani polovinu dostupných funkcí.

¹Portable Document Format je souborový formát pro ukládání dokumentů.

²JavaScript Object Notation je způsob zápisu dat.

O rok později jsme měli možnost, v rámci předmětu rétorika, si práci s Power-Pointem zopakovat, když jsme dostali za úkol vytvořit prezentaci na dané téma. Při prezentování studentů jsem byl překvapen, že podobně jako učitelé, nikdo nepoužívá složitější funkce. Naopak jsem tady narazil ještě na další problém, kterému se učitelé úspěšně vyhýbali. Jelikož se prezentovalo na školním počítači, bylo nutné studentem vytvořenou prezentaci na daný stroj nějakým způsobem dostat. Ti technicky zdatnější použili na dopravu datové médium, ostatní využili možnosti poslat sami sobě e-mail obsahující prezentaci, a to i tehdy, když jejich e-mailová služba nabízela cloudové úložiště³. Potom zde byla skupina lidí, kteří si prezentaci zapomněli doma a díky tomu byli ohodnoceni známkou nedostatečně. V té chvíli se mi v hlavě začala rodit myšlenka, která by řešila všechny základní potřeby běžného přednášejícího, a to zdarma.

Ani ve čtvrtém ročníku jsme se prezentacím nevyhnuli. Prezentování probíhalo téměř stejně jako rok minulý, až na jednoho spolužáka, jehož prezentace se mi zdála velmi profesionální. Při konverzaci mi prozradil, že se jedná o webovou aplikaci Prezi (více v podsekcí 2.2) a nabídl mi, že mi ji představí. Byl jsem překvapen, že něco takového o čem jsem snil, již existuje. Stále se mi však jevila trochu složitě a vypadala, že obsahuje navíc několik nadbytečných funkcí. Jinak mi ale připadala bez chyby. Rozhodl jsem se aplikaci vyzkoušet. Byla chráněná registrací, která mě zpočátku lehce odradila, ale ne tak, abych to nepřekonal. Chvilí jsem ji zkoušel, ale jelikož jsem neměl žádnou příležitost něco prezentovat, brzy jsem toho nechal.

Po roce jsem se rozhodl Prezi při jiné příležitosti znova použít. Protože jsem ji však už skoro rok neužíval, nepamatoval jsem si e-mail ani heslo, a tak nezbývalo nic jiného, než si vytvořit účet nový. Postupem času jsem zjistil, že toto zapomenutí údajů není tak neobvyklé ani u ostatních uživatelů. Uživatel si vytvoří pár prezentací, zapomene heslo a pak jeho výtvořů jen zbytečně zabírají místo. Zabezpečení registrací se mi zdálo zbytečné, zvláště proto, že Prezi nabízí přímo variantu pro studenty, kteří často používají k registraci školní e-maily, ty však ale již po skončení studia nemohou použít.

K přenositelnosti, jednoduchosti a potřebě nic neplatit jsem se po mých zkušenostech rozhodl přidat čtvrtou, poslední vlastnost – vytváření prezentace bez registrace. Žijeme v době, kdy můžeme bez registrace používat ve webovém prohlížeči malování, textový blok, hry nebo dokonce i jednoduchou verzi Photoshopu⁴, tak proč ne software na tvorbu prezentací? Rozhodl jsem se vytvořit aplikaci, která bude nabízet z mojí zkušenosti jen nejčastěji používané funkce, bude zdarma, bez registrace a dostupná zadáním URL adresy⁵.

³Služba, která umožňuje ukládat data tak, že se přenesou přes síť do úložného systému.

⁴<https://www.photopia.com/>

⁵Adresa určující umístění na internetu.

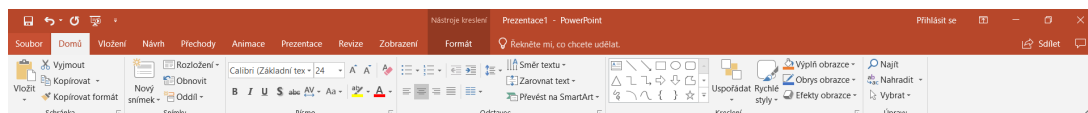
2 Existující aplikace

Existuje celá řada aplikací umožňující vytvářet prezentace na počítači. Většinou se jedná o podobné aplikace lišící se jen v několika málo detailech. Daly by se rozdělit na dvě hlavní skupiny - placené a zdarma. Při zkoušení jsem se zaměřil na ty, co jsem mohl vyzkoušet zdarma. Ve svém pátrání jsem nenarazil na žádnou, která by nevyžadovala registraci, předpokládám tedy, že Slidly je jediná. V této kapitole popíšu a představím výhody a nevýhody, mnou vybraných tří hlavních představitelů. A srovnám s nimi moji aplikaci.

2.1 PowerPoint

PowerPoint [1] je velmi populární desktopovou⁶ aplikací od společnosti Microsoft. Díky této popularitě se v mnoha případech stal synonymem pro pojem počítačová prezentace a dokonce se vyučuje i na některých školách.

Hlavní výhodou, ale částečně i nevýhodou, je velké množství funkcí, které můžeme použít. Kromě těch základních je to například tvorba grafů, tabulek, vkládání matematických výrazů, časovače či animační přechody. Nabízí také řadu motivů pro atraktivní vzhled prezentací. Jelikož se jedná o desktopovou aplikaci, běží rychle a při dlouhodobém používání jsem nenarazil na žádnou chybu. Pro aktivní uživatele využívající balík služeb Office 365 a pracující v týmu, umožní spoluvytváření v reálném čase. Dále nabízí i možnost generování do PDF a dostupnost bez internetového připojení. Microsoft také poskytuje online verzi PowerPointu v prohlížeči. K její používání je však nutná registrace a bohužel zde ztrácí část rychlosti a nabízí méně funkcí než její desktopový protějšek.



Obrázek 1: Nabídka funkcí karty Domů.

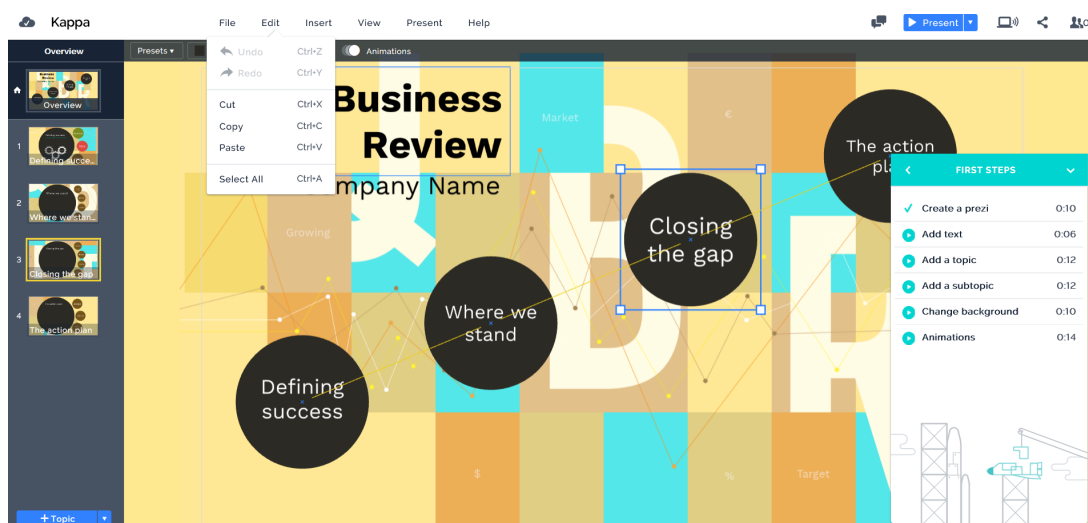
Nevýhodou je pak hlavně cena, kterou je nutné zaplatit. Ta je zvýšena i díky tomu, že se PowerPoint prodává společně v balíku s ostatními kancelářskými aplikacemi Microsoft Office. Zadarmo ho k používání nabízí některé školy, díky čemuž jsem jej mohl vyzkoušet i já. Zvláště uživatelé, kteří nepracovali s žádnou aplikací ze zmíněného balíku, mohou trpět zmateností z nabízených možností, které, jak je částečně vidět na obrázku číslo 1, jsou rozházeny do jednotlivých karet. Pocit zmatenosti úplně nezmizí ani u pokročilejších uživatelů, kteří mohou mít stále problém najít méně využívané funkce. Tento problém je dán velkým množstvím položek, jehož výsledkem je to, že často používané funkce leží ve stejných místech jako ty méně používané. Takže většinu času jen zhoršují přehlednost a ovládání. Také zde chybí podpora pro Linux. Pro uživatele, kteří občas vytvoří jednoduchou prezentaci se mi tak PowerPoint jeví zbytečný.

⁶Aplikace, která běží díky operačnímu systému na našem počítači, startuje z pevného disku.

2.2 Prezi

Prezi [2] patří mezi hlavní představitele webových aplikací pro tvorbu prezentací. Na svých stránkách se chlubí větou, která nás informuje o tom, že výzkumníci z Harvardu považují Prezi za více poutavé, přesvědčivé a efektivní než PowerPoint.

Za výhody této webové aplikace považují o něco méně složitější nabídku funkcí a přenositelnost. Hlavním plusem, kterým tato aplikace předčí ostatní, jsou však profesionální šablony a animace, které z vytvářených prezentací vytváří krásná umělecká díla bez větších problémů. Aplikace nepotřebuje instalovat podpůrné programy a je, alespoň v základní verzi, zdarma dostupná v internetovém prohlížeči.



Obrázek 2: Nově vytvořená prezentace v Prezi.

Hlavní nevýhodou je omezení verze zdarma, které například schází generování do PDF, nemožnost prezentovat offline nebo kontrola soukromí. A ačkoliv neobsahuje celou další řadu funkcí, stále je nutná registrace. Jiné nevýhody jsou vidět na obrázku číslo 2. Aplikace zpočátku působí velmi složitě. Vlevo vidíme panel obsahující náhled celé, oddálené prezentace a pod ním automaticky vytvořené snímky, které můžou nového uživatele lehce zmást. I když se zde vpravo nachází nápověda, nabízí i rady k položkám, ke kterým se uživatel ještě nedostal, což může působit lehce chaoticky. Navíc tato nápověda zavádí pojmy „topic“ a „subtopic“, což jsou pro uživatele přecházející z podobných aplikací, nové výrazy, jejichž koncept musí pochopit. To pak může zpomalit start používání. Také zde můžeme vidět o něco přehlednější menu, než v případě PowerPointu. To však trpí problémem skrytých funkcí. Uživatel je tedy nucen otevírat nabídky i pro často používané funkce jako je například zpět a vpřed. Podobně jako v předešlém případě není aplikace vhodná pro občasné tvůrce prezentací.

2.3 Prezentace Google

Prezentace Google [3] je další webovou aplikací, která je často používaná hlavně díky tomu, že je spojená s velmi populárními službami Googlu.

Výhodou této webové aplikace je rozumná nabídka funkcí s tím, že nejčastěji používané jsou přímo přístupné z docela přehledného menu. I když je k používání nutná registrace, plná verze aplikace je zdarma. Uživatel jistě uvítá i možnost prezentaci převést do několika typů souborů včetně PDF. Aplikace sama o sobě nabízí několik motivů a existuje i možnost motiv importovat. Jako jiné webové kancelářské aplikace od Google nevyžaduje instalování podpůrných aplikací a nabízí jednoduché sdílení a práci ve více lidech.

Nevýhodu vidím v tom, že pokud chce uživatel používat třeba jen jednu danou webovou aplikaci, stále je nutné se zaregistrovat pro celý Google. Tedy i e-mail, což je pro občasné vytváření prezentací zcela zbytečné. Problém je také sdílení prezentací bez přihlášení, i když je možné přistoupit k prezentaci bez přihlášení pomocí adresy. Kvůli její složitosti pak této možnosti uživatel často nevyužije.

2.4 Srovnání

Slidly se snaží držet čtyř zvolených vlastností - přenositelnost, jednoduchost, měla by být zdarma a bez registrace. Také se vyhýbá řadě problémům již zmíněných aplikací. Ale nebojí se inspirovat v některých částech jako je například ovládání, které bylo ve výše zmíněných aplikacích téměř totožné a je tedy prověřené.

Výhodou je například nabídka základních jednoduchých funkcí, které jsou umístěné v přehledném menu, jenž je rozdělené podle toho, s čím zrovna chce uživatel pracovat. Žádná jeho část není skrytá, uživatel tedy nemusí proklikávat a hledat jednotlivé funkce. Nevyžaduje instalaci podpůrných programů. Je celá zdarma a dostupná v prohlížeči s možností vygenerovat slajdy do PDF. Důležitá je i možnost privátní prezentace, která je chráněná heslem. Dále nabízí koncept práce s konzolí. Ta umožní zrychlit některé akce a do budoucna má potenciál řadu věcí zautomatizovat. Navíc, i když je to webová aplikace, umožňuje, aby si sám uživatel přidával různé fonty. Hlavním plusem, oproti ostatním, je však vytvoření prezentace bez registrace. Tato prezentace je pak dostupná na uživatelem zadané adrese, díky čemuž odpadá problém posílání složitých odkazů na e-maily či jinde. Slidly je tak velmi vhodné pro uživatele, kteří tvoří prezentace občasně.

Sebekriticky musím napsat, že má ale i pár nevýhod – absenci motivů a více-uživatelské práce, menší podporu prohlížečů a větší šanci na výskyt chyby. Tyto problémy jsou dány tím, že na dříve zmíněných aplikacích pracují celé týmy a fungují už řadu let. Slidly vytvářím jenom já a celkově funguje pouze krátkou dobu. Možnost víceuživatelské práce jsem nemohl implementovat hlavně kvůli chybějící registraci, která pro mě byla důležitější. Ostatní problémy se však již nyní snažím řešit. Vytváření motivů je možné pomocí konzole (viz podsektce 5.8). Podpora a chybovost se pak zlepší postupným používáním a vývojem aplikace.

3 Použité technologie

Pro webové aplikace je výběr technologií klíčový. Existují a stále vznikají nové možnosti, jak řešit problémy různého typu. V této kapitole představím mnou vybrané technologie a důvody, které jsem měl pro jejich použití.

3.1 JavaScript

JavaScript [4] je interpretovaný, dynamicky typovaný, objektově orientovaný programovací jazyk, který se často používá na webu pro ovládání interaktivních prvků stránky. Jeho možnosti jsou však daleko větší. Na rozdíl od typických objektově orientovaných jazyků, které používají třídy, JavaScript využívá prototypy. Třídy, které ale spíše slouží jenom jako syntaktický cukr⁷, byly představeny až ve verzi ECMAScript⁸ 2015 [5]. Jelikož je JavaScript interpretovaný prohlížeči, nelze se spolehnout na to, že uživatel bude používat prohlížeč, který dokáže využívat všechny funkce JavaScriptu. Tento problém částečně řeší například Babel⁹, který jsem při práci také použil. Jazyk dále nabízí možnost využívat různá paradigmatata programování – imperativní, objektové i funkcionální.

V dnešní době se nepoužívá jenom pro upravování zobrazené stránky, ale také ke komunikaci se serverem [6] nebo na serveru samotném (viz 3.4.) To společně s podporou všech hlavních prohlížečů dělá z JavaScriptu v podstatě nutnost pro vytváření webových aplikací, bez které jsem nemohl práci udělat.

3.2 HTML 5

HTML [7] je zkratkou pro HyperText Markup Language. Jedná se o značkovací jazyk používaný pro definování struktury a sémantiky webových stránek. Zdrojový kód s příponou .html se skládá ze značek, kterým se říká tagy. Těmito tagům lze přiřazovat atributy, jenž, mimo mnoho jiného, zjednodušují práci JavaScriptu a CSS (více 3.3). Je interpretovatelný webovými prohlížeči, které však často způsobují problémy. Jelikož je HTML stále vyvíjen, vývojáři prohlížečů musí nové prvky pořád implementovat. Často tedy nastává situace, kdy nějaký prohlížeč nepodporuje daný element jazyka. O standardy se stará W3C¹⁰. HTML 5 je pak hlavní označení pro nejnovější verze.

HyperText Markup Language je klíčovým prvkem každé webové stránky, proto bylo nutné ho při mé práci využít. V této aplikaci nově představené strukturální prvky HTML 5 používám převážně jen na úvodní straně, kde jsou velmi důležité. V částech webové aplikace je kód generovaný JavaScriptem, struktura nabízená novými HTML 5 prvky je pak těžce nasaditelná tak, aby dávala smysl a mnohdy je i zbytečná.

⁷Část syntaxe programovacího jazyka, jejíž účelem je usnadnit zápis.

⁸JavaScript je implementací ECMAScriptu.

⁹JavaScriptový kompilátor <https://babeljs.io/>.

¹⁰Konsorcium W3C (World Wide Web Consortium) <https://www.w3.org/>.

3.3 CSS 3

CSS [8] je zkratkou pro Cascading Style Sheets. Jde o jazyk, který popisuje způsob, jakým jsou HTML elementy stylovány, tedy zobrazeny uživateli. Styly se zapisují pomocí CSS pravidel skládající se ze selektoru určující HTML elementy, vlastnosti a jejich hodnot. V této práci používám CSS nejen napojením souboru s příponou .css do HTML, ale také v JSX (více 3.6), kde se ke stylování používá JavaScriptový objekt, jehož vlastnosti jsou rovny vlastnostem CSS s tím rozdílem, že jména vlastností obsahující pomlčku přepíšeme do velbloudí notace¹¹. Jejich hodnotou je pak řetězec.

Stylování pro moji aplikaci bylo klíčové, ačkoliv často stránky typu diskuzních fór, internetových encyklopedií nebo třeba portfoliích mohou být funkční i bez stylu. Slidly, jakožto webová aplikace pro tvorbu prezentací, je na stylech závislá, jelikož jednotlivé prvky musí zobrazovat informace o velikosti, barvě a poloze. CSS je tak pro mě stejně nezbytné jako HTML nebo JavaScript.

3.4 Node.js

Node.js [9] je asynchronní, událostmi řízené běhové prostředí pro JavaScript, využívající vysoce výkonný, v C++ napsaný, V8 engine od Googlu, který využívá například i Google Chrome. Node.js se používá na straně serveru. Na rozdíl, od často se vyskytující varianty, kdy je využíváno několika vláken, při psaní kódu Node.js se používá pouze jedno vlákno se smyčkou událostí. Ta umožňuje provádět neblokující vstupně/výstupní operace, přestože JavaScript je jen jednovláknový. Tím, že předává operace jádru systému, které většinou vícevláknové je. Když je daná operace hotová, jádro informuje Node.js a díky tomu může být callback přidán do fronty k vykonání [10]. Tyto vlastnosti dovolují v Node.js tvořit i škálovatelné systémy. Prostedí Node.js navíc umožňuje využívat řadu modulů, které se dají jednoduše stáhnout pomocí npm¹². V mém případě jsou to například express (viz 3.5), cookies-parser, body-parser, express-fileupload, cors nebo mongoose.

Ačkoliv se na straně serveru nabízí i řada jiných možností, zvolil jsem si tuto. A to z důvodu škálovatelnosti, nezávislosti na platformě, vhodnosti pro jednostránkové aplikace, práce s vlákny, jednoduchého vytvoření API¹³, pohodlné práce s JSON a také kvůli možnosti použít jeden jazyk na serveru i klientovi. Do budoucna je také vhodný například i pro registrace a případnou implementaci víceuživatelské práce s jednou prezentací.

¹¹Způsob psaní, kde nejsou jednotlivá slova oddělena mezerami, navíc slova začínají velkým písmenem.

¹²Správce balíčků pro JavaScript <https://www.npmjs.com/>.

¹³Rozhraní pro programování aplikací.

3.5 Express

Express [11] je rychlý, flexibilní framework pro tvorbu Node.js aplikací. Poskytuje řadu funkcí nejen pro webové, ale i mobilní aplikace. Slouží pro usnadnění rychlého vývoje. Express je postaven na principu „middleware“ funkcí. Což jsou funkce, které umožňují reagovat na http a https požadavky, mají přístup k „request“ a „response“ objektu a následující „middleware“ funkci v „request-response“ cyklu aplikace, které se říká „next“. Tyto funkce umožňují vykonávat kód, měnit „request“ a „response“ objekty, ukončit „request-response“ cyklus nebo zavolat další „middleware“ funkci v pořadí [12].

Tento framework jsem si zvolil ze tří hlavních důvodů - jednoduchého vytvoření serveru, „middleware“ funkcí a dalšího důležitého prvku - směrovačů. Pomocí nich jsem rozdělil kód na jednotlivé části. Směrovače mi umožní jednoduše přidat zcela novou funkcionalitu do již vytvořeného serveru. Tento krok se ukázal jako vhodný při implementaci služby pro převod obrázku (viz 4.5). Často se také používá pro renderování HTML stránek, kdy předává šablonám data, ale ve své práci používám Express jen pro směrování a definování „middlewaru“.

3.6 React

React [14] je JavaScriptová knihovna, vyvinutá společností Facebook, která se převážně používá pro vytváření uživatelských rozhraní. Umožňuje definovat znovupoužitelné zapouzdřené komponenty, které mají vlastní stav a funkce. Výsledkem jejich kombinací jsou komplexní uživatelská prostředí. Logika těchto komponent je psána v JavaScriptu, díky čemuž jsou stavy dat oddělené od DOMu¹⁴. Nabízí také virtuální DOM, kdy je reprezentace uživatelského rozhraní uložena v paměti a do reálného DOMu jsou efektivně promítnuty jen důležité změny, které zbytečně aplikaci nezpomalují. Komponenty implementují funkci `render`, v nichž se používá jazyk podobný HTML nazývaný JSX. Nejedná se však o HTML, nýbrž o syntaktické rozšíření JavaScriptu. Používá se převážně kvůli čitelnosti a do JavaScriptu je posléze překládán pomocí Babelu. Místo JSX je možno použít funkci `React.createElement()` ;

Pro React jsem se rozhodl hlavně z důvodu znovupoužitelných komponent, které usnadní nejen možnost jednoduše přidat nové prvky do aplikace, ale také použít mnou vytvořené komponenty i v dalších projektech. Příkladem může být komponenta `Image` ve zdrojovém kódu číslo 1. Ta zobrazí obrázek rozmátnutý do zadané velikosti. Tohoto chování využívám, když ji vkládám do `ResizeContainer`, což je komponenta, která dokáže měnit velikost podle táhnutí myši a obrázek tím tak mění i svou velikost. `Image` by však šlo využít i v grafických editorech kde by uživatel mohl obrázek libovolně nastavovat.

¹⁴Document Object Model - reprezentace HTML dokument.

```

1 export default class Image extends React.Component {
2   render() {
3     return(
4       <div
5         style={{
6           background: 'url("' + this.props.src + '")',
7           backgroundRepeat: 'no-repeat',
8           backgroundSize: '100% 100%',
9           width: this.props.w,
10          height: this.props.h,
11          border: this.props.border
12        }}>
13     </div>
14   );
15 }
16 }

```

Zdrojový kód 1: Definice komponenty Image.

3.7 Draft.js

Draft.js [15] je JavaScriptový framework pro tvorbu textových editorů v Reactu, který zajišťuje funkcionalitu napříč prohlížeči, stylování textu, vkládání některých multimédií a další užitečné prvky. Důležitou částí editorů v Draft.js je „Editor State“, využívající immutable-js, který se drží přístupu, že jakmile jsou data vytvořena nemůžou být změněna [16]. Tento přístup provází Draft.js i jinde.

Pro Draft.js jsem se rozhodl, protože na rozdíl od klasického HTML elementu `textarea`, se kterým React nedovoluje lehce provádět některé akce nutné při tvoření textového editoru s formátováním, Draft.js je přímo vytvořen pro práci s Reactem. Navíc nabízí řadu funkcí, pro jejichž zprovoznění je však nutné napsat docela velké množství kódu. Při používání jsem narazil i na problémy. Například při definování různých vlastností písma nebo se styly odrážek doporučené Draft.js, které při změně velikosti nefungovaly správně. Tyto problémy bylo nutné obcházet. Co však je výhodou Draft.js je jednoduchá možnost uložení aktuálního stavu a jeho opětovného načtení. Tento stav je vytvořen jako JSON, díky čemu je možné ho využít na platformách, kde Draft.js není. Draft.js také nabízí již hotová řešení ve formě pluginů, kterých jsem se rozhodl nevyužít, pro lepší pochopení toho jak funguje.

3.8 MongoDB

MongoDB [17] patří mezi hlavní představitele NoSQL¹⁵ databází. Je to vyspělý, škálovatelný, databázový systém. Jedná se o databázi, ve které je řádek nahrazen strukturovaným dokumentem. V případě MongoDB je tímto dokumentem

¹⁵Koncept databáze, kde se nepoužívají tabulková schémata jako například u relačních databází.

BSON (binární JSON), který uchovává v kolekcích. MongoDB umožňuje provádět dotazy přímo na data, a tím zachovává velkou část možností, kterou nabízí například SQL¹⁶.

Pro MongoDB jsem se rozhodl, jelikož se jedná o skvělý doplněk k Node.js. Data v MongoDB mají flexibilní schéma, což se může hodit nejen pro postupný vývoj, ale také pro případné rozšíření aplikace. Tyto vlastnosti, společně s tím, že je zdarma a k uchovávání dat potřebují pouze jednu kolekci s jedním typem záznamu, vytvořily perfektní volbu.

3.9 Mongoose

Mongoose [18] je balíček poskytující objektové modelování pro Node.js. Díky němu se můžeme připojit k MongoDB, což umožní používat validace, logiku a dotazy, které MongoDB samo nenabízí. Mongoose používá takzvaná schémata. V mém případě mám pouze jedno schéma, které je možno vidět ve zdrojovém kódu číslo 2. Ke každému schématu existuje kolekce a schéma jí určuje, jak budou vypadat dokumenty v ní obsažené. Každý klíč ve schématu definuje vlastnost dokumentu. Ze schématu se vytvoří konstruktor – model. Instance modelu jsou poté nazývány dokumenty. Ty reprezentují dokumenty uložené v MongoDB.

Důvodem pro použití je hlavně to, že sebou přináší řád, který MongoDB chybí. Zároveň mě zaujaly jeho dotazy a také například časová známka, kterou mohu využít při odstraňování prezentací po 30 dnech. To u mě rozhodlo Mongoose použít.

```
1 const PresentationSchema = new Schema({
2   _id: {
3     type: String,
4   },
5   password: {
6     type: String,
7   },
8   data: {
9     type: String,
10  },
11  public: {
12    type: Boolean,
13    default: true
14  },
15 }, { timestamps: true });
```

Zdrojový kód 2: Definice schématu prezentace.

¹⁶Dotazovací jazyk, který je používán pro práci s daty v relačních databázích.

3.10 html2canvas, jspdf

Kombinací knihoven `html2canvas` [19] a `jspdf` [20] můžeme částečně řešit problém generování HTML obsahu do PDF. V současné době totiž prohlížeče nenabízí téměř žádné řešení pro generování webových stránek do PDF a knihovnu, řešící tento problém korektně, se mi nepodařilo najít.

Ačkoliv `jspdf` sice nabízí řadu možností generování PDF, v době psaní mé práce bohužel neumožňovala přímo vytvořit PDF z HTML a CSS. Bylo by nutné napsat vlastní překladač, který by přeložil mnou ukládaná data na data akceptovatelné `jspdf`. Toto řešení jsem zavrhl z důvodu složitosti a hlavně závislosti dat na knihovně `Draft.js` a `jspdf`. Při změně formátu dat, ať už `jspdf` nebo `Draft.js`, by bylo nutné překladač přepsat.

Knihovnu `jspdf` je tedy nutné zkombinovat s knihovnou `html2canvas` (viz 4.6), která dokáže převést HTML element společně s jeho stylem na obrázek, který lze následně použít při generování PDF. `Html2canvas` s sebou přináší řadu problémů. V době psaní práce neumožňovala korektně převést všechny vlastnosti CSS 3, čímž mě připravila o možnost využít například rotace boxů. Na knihovně se stále pracuje, takže tato překážka by měla být v budoucnu vyřešena. Další problém, který to přineslo je práce s textem – jelikož je PDF soubor tvořen obrázky, není možné text v něm zkopírovat nebo na něj kliknout a nelze tedy používat ani odkazy. Tyto problémy však nenarušují mé základní myšlenky o aplikaci, které se zaměřovaly hlavně na přenositelnost a jednoduchost. Proto jsem se rozhodl upřednostnit generování do PDF před prací s textem a používáním efektů.

3.11 Web Font Loader

`Web Font Loader` [21] je knihovna, která umožní načítat fonty i při běhu aplikace. Nabízí možnost načítání z několika zdrojů včetně `Google Fonts`¹⁷, kterého jsem v projektu využil.

K tomuto kroku jsem přistoupil, protože se mi zdálo nemožné podchytit všechny styly písma, které by uživatel chtěl využít. Uživatel tak není závislý na mnou vybraných webových fontech, které by jeho platforma ani nemusela podporovat. Navíc ohromný výběr fontů je velmi důležitý při vytváření atraktivních prezentací.

3.12 Font Awesome

Mimo technologie tvořící funkční jádro jsem také použil sadu ikon a nástrojů `Font Awesome` [22] v její páté verzi. Obsahuje několik stovek vektorových ikon, kterými zkrášluji jednotlivé položky v aplikaci. Položkám tak dodám profesionálnější vzhled a navíc ikony usnadní uživateli pochopit co daná akce dělá.

¹⁷<https://fonts.google.com/>

4 Programátorské dokumentace

V této kapitole popíši základní rozdělení práce do adresářů, návrh databáze, zabezpečení a přístupy, které jsem zvolil při řešení aplikace.

4.1 Adresářová struktura

Samotná práce se skládá ze dvou částí, Node.js serveru a React aplikace. Toto rozdělení jsem zachoval i v adresářové struktuře, kterou jsem rozdělil na dvě části server a aplikace. V adresáři server nalezneme kromě souborů s kódy, také složku `public`, obsahující obrázky nahrávané uživateli. Rozložení složek a souborů částečně vychází z kostry aplikace vzniklé při vygenerování¹⁸ nového Express projektu. Rozdílem, oproti této struktuře, je však absence adresářů `bin` a `views`, které souvisí s vytvářením stránek. Od nich jsem upustil, jelikož Express využívám jen pro směrování a middleware.

Protože React neposkytuje přesně danou strukturu pro dodržení [23], použil jsem podobný přístup jako v případě Expressu. Vycházel jsem ze struktury nově vygenerované¹⁹ aplikace React. Struktura aplikace je tvořena složkami `public`, která obsahuje HTML, CSS soubory a složkou `src`, která obsahuje kód samotné aplikace rozdělený na komponenty, soubory s užitečným kódem a soubor `config.js` obsahující důležité adresy pro fungování aplikace. Tento soubor je možné přepsat v případě, že se změní hosting nebo server.

V obou adresářích se také nachází soubor `package.json`. Ten obsahuje informace o projektech, závislostech, ale také o skriptech. Je velmi důležitý pro lokální spuštění aplikace (viz příloha A).

4.2 Zabezpečení

Aplikace nemá za cíl zatěžovat uživatele registrací ani zadáváním osobních údajů, dokonce ani e-mailu. Díky tomu je motivace útočit menší než v případě jiných podobných projektů. Při vývoji, ale i tak nastal problém, jak data ochránit, nejen proti útočníkům, ale i proti ostatním uživatelům. Potřeba zabezpečení je pouze v případě prezentace samotné, a to jak k jejímu čtení (tato ochrana je volitelná), tak k její editaci.

Problém ochrany jsem vyřešil použitím hesla k prezentaci, které je uloženo v databázi. Ačkoliv heslo nechrání důležitá osobní data, stále je potřeba ho za-hashovat²⁰. K jeho zabezpečení je využito modulu `bcrypt` [24]. Ten využívá stejnojmennou hashovací funkci, která je založená na šifře Blowfish²¹ a navíc zahrnuje sama o sobě sůl²², která ještě zmenší šanci prolomení hesla.

¹⁸<https://expressjs.com/en/starter/generator.html>

¹⁹<https://github.com/facebook/create-react-app>

²⁰Místo uložení v čitelné podobě se ukládá pouze jeho otisk.

²¹Symetrická bloková šifra.

²²Náhodné bity přidané k heslu.

Důležité vstupy od uživatele jsou kontrolovány na serveru a při vypisování pomocí JSX se předchází XSS²³. Díky použití MongoDB také odpadá problém s tradičními SQL injekcemi²⁴. Navíc, protože nevyužívám přihlašování a ani session, odpadají některé další bezpečnostní problémy.

4.3 Databáze

Data je nutné perzistentně uložit. Jednotlivé záznamy, jako je například 3, které ukládám, fungují jako samostatné jednotky, nejsou závislé na jiných záznamech či kolekcích a ani spolu nesouvisí. Relační databáze by zde tedy nevyužila svůj plný potenciál, jelikož zde nebudu vytvářet relace mezi daty různých tabulek. Z tohoto důvodu, dobré spolupráce s Node.js a jednoduchosti, jsem zvolil MongoDB. Do kolekce „presentations“ v databázi ukládám pouze jeden typ dokumentu. Ten představuje prezentaci. Tento dokument pak vychází ze schématu stanoveným Mongoose.

Pro konkrétní databázi využívám cloudovou službu mLab²⁵. Tato služba nabízí verzi zdarma a hlavně dostupnost, která je jedním z cílů mé práce. Uživatel, který si zprovozní aplikaci lokálně, tak bude moci používat stejná data jako uživatel využívající serveru. Tuto službu lze však snadno nahradit i lokální, či jinou MongoDB databází. Tím, že se v souboru `app.js` v adresáři `server` změní hodnota argumentu funkce `mongoose.connect('nova_db');`.

```
1 {
2   "_id": "record",
3   "updatedAt": {
4     "$date": "2018-07-22T13:41:59.822Z"
5   },
6   "createdAt": {
7     "$date": "2018-07-22T13:41:59.822Z"
8   },
9   "data": {"width":1920,"height":1080,"slides":[],"fonts":
10     :[]},
11   "password": "\$2a\$10\$g.BdW4U79mOYjq1WlUMti.LmZiDpwqN40.6/
12     fGUSQHiqbVVAOUWHW",
13   "public": true,
14   "__v": 0
15 }
```

Zdrojový kód 3: Uložený záznam, který neobsahuje slajdy.

²³Cross-site scripting je útok, při kterém útočník podstrčí skript, který je pak zobrazen ostatním uživatelům.

²⁴Typ napadení databáze přes neošetřený vstup, kde útočník upraví SQL dotaz.

²⁵<https://mlab.com/>

4.4 REST API

Serverová část musí být připravena i pro budoucí rozšiřitelnost. V případě, že budu já nebo někdo jiný chtít službu rozšířit i na jiné platformy. Rozhodl jsem se předejít možným problémům tím, že už teď vytvořím veřejné rozhraní, které bude srozumitelné jak pro aplikace, tak pro i programátory, kteří pak nemusí vědět jak server vnitřně funguje, aby novou aplikaci vytvořili. Toto rozhraní se nachází na URL adrese serveru, konkrétně „/api“.

Jako rozhraní jsem zvolil dnes velmi populární REST API, což je označení architektury, umožňující pomocí metod provádět operace se zdroji, které mají vlastní URI²⁶. Metody RESTu jsou známy pod označením CRUD, tedy Create, Read, Update a Delete. Konkrétně se pro ně využívají http metody POST, GET, PUT a DELETE. Obdržená data jsou pak nejčastěji ve formátu XML nebo JSON. [25]

Kvůli použití MongoDB a JavaScriptu jak na straně serveru, tak na straně klienta, jsem se rozhodl pro JSON. Pro konkrétní odpovědi, které slouží k informaci, jsem pak zvolil vlastní reprezentaci skládající se z objektu (například zdrojový kód 4) s vlastnostmi určující úspěch, typ a text. Tato odpověď je pak přímo napojitelná na mou komponentu Message, která ji potom zobrazuje.

```
1  savesuccess: {
2      successfully: true,
3      type: 'success',
4      message: 'Úspěšně uloženo!'
5  }
```

Zdrojový kód 4: Odpověď při úspěšném uložení prezentace.

4.5 Služba pro převod obrázků

Kromě problémů knihovny html2canvas s některými vlastnostmi CSS 3, je zde také potíž s obrázky. Knihovna nezachytí obrázky, které nesplňují takzvanou „same-origin policy“²⁷. Je tedy nutné buď použít proxy anebo umístit obrázek pod stejný zdroj - origin, který je definován jako kombinace URI schématu, jména hostitele a portu [26]. Tato řešení se mi, z důvodu práce bez internetového připojení, složitosti a bezpečnosti, zdála nevhodná. Rozhodl jsem se pro možnost využití Base64²⁸, se kterými html2canvas tolik problémů nemá. Na serveru jsem tedy, pro požadavky obsahující adresu obrázku a směřující na „/base64“, vytvořil směrovač, který daný obrázek převede do kódování Base64. Toto řešení má také řadu nevýhod. Mezi hlavními je ta, že je třeba kontaktovat server, který je těmito požadavky zbytečně vytížen a také musí přistupovat k adresám obrázku.

²⁶Uniform Resource Identifier – řetězec se strukturou, který slouží k specifikaci zdroje.

²⁷Bezpečnostní model používaný webovými aplikacemi.

²⁸Kódování, které převádí binární data na znaky.

Tyto problémy mohou mít za následek pomalejší načtení obrázků v PDF editoru či možné problémy při převádění způsobené neschopností zajistit, že obsah adresy obrázku bude k dispozici. Tyto žádosti však probíhají pouze jednou a až v PDF editoru a samotná tvorba slajdů a jejich prezentace jimi narušena není. Převody provádím na serveru, kvůli lepší kontrole, výsledkům a pro případ, že by se v budoucnu implementovala jiná aplikace, ve které by byl podobný problém.

4.6 PdfConverter

Jak jsem se v podsekcí 3.10 zmínil, pro generování prezentací do PDF souborů je nutné zkombinovat knihovny `html2canvas` a `jspdf`. Vytvořil jsem tedy vlastní malou knihovnu, která slouží pro tento účel. Nazval jsem ji `PdfConverter` a jejím úkolem je zpracovat a převést jednotlivé slajdy do PDF. Kromě toho je připravena i pro případ špatně převedeného obrázku.

4.7 Vlastní komponenty

Projekty vytvořené pomocí Reactu jsou složeny z komponent, kterých jsem sám při práci na aplikaci vytvořil celou řadu. Některé z nich jsou závislé na tomto projektu, ale většina je připravená být využita i někde jinde. Velké množství komponent jsou pak propojeny v souboru `Presentation.js` v adresáři `src`.

Pro práci bylo také nutné vytvořit vlastní box, se kterým by bylo možné pohybovat a měnit jeho velikost nezávisle na jeho obsahu, u kterého by bylo nutné zajistit, aby se choval korektně i při změně oddálení slajdu. Dále komponentu, jenž by podle objektu (například zdrojový kód 5) dokázala vytvořit vyskakovací okno s požadovanou funkcí, ale také například i komponentu pro náhledy slajdů. Ta by měla chytře překreslovat pouze měněnou část a to pouze v případě dokončení změn, čímž se vyhne zbytečně častému překreslování, které by aplikaci zpomalilo. Tyto a další komponenty jsou umístěné v `aplikace\src\Components`.

```
1 {
2   title: 'Uložení prezentance',
3   subtitlePw: 'Zadejte heslo',
4   commentPw: 'Heslo slouží k uložení provedených změn',
5   elementPw: { type: 'input', options: 'password', value: '' },
6   subtitlePublic: 'Veřejná',
7   commentPublic: 'Veřejnost umožní lidem vidět vaši prezentaci bez
      hesla',
8   elementPublic: { type: 'input', options: 'checkbox', value:
      isPublic },
9   commentBtns: '',
10  okBtn: okBtn,
11  closeBtn: closeBtn
12 }
```

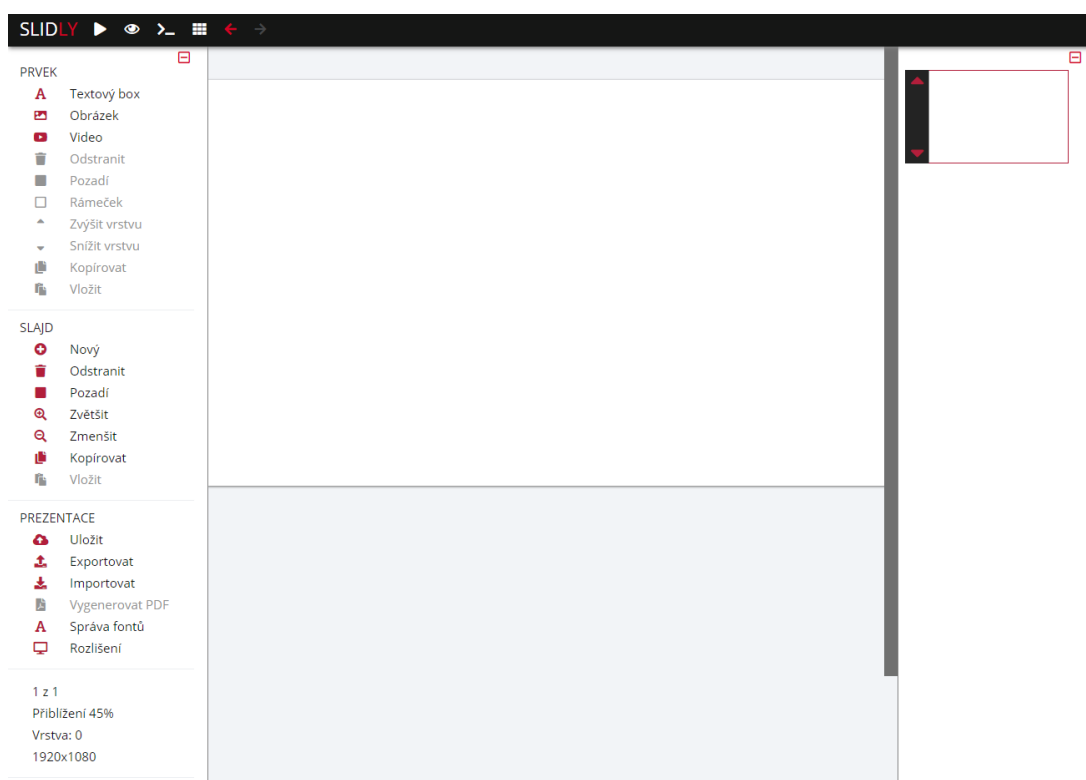
Zdrojový kód 5: Objekt, podle kterého se vygeneruje okno uložení.

5 Uživatelská dokumentace

Tato část slouží jako příručka pro uživatele. Popisuje funkce a základní kroky pro vytvoření prezentace. Jelikož se jedná o webovou aplikaci, k použití popsaných kroků stačí jen internetový prohlížeč. Doporučeným prohlížečem pro používání jsou novější verze Google Chrome, více informací v příloze C.

5.1 Rozložení aplikace

Editor se skládá ze 4 částí. Na levé straně obrazovky se nachází menu nabízející většinu funkcí. Je členěno na 3 části, které oddělují položky menu podle objektu, se kterými chce uživatel pracovat. Kromě oddílů s funkcemi zde nalezneme i informace o aktuálním slajdu a boxu. Většinu obrazovky zabírá kontejner obsahující slajd, jenž uživatel tvoří. Vpravo je náhled již vytvořených slajdů. Ten zobrazuje i prováděné změny. Tyto změny jsou vykresleny až pod podmínkou, že za poslední půl vteřiny se jejich počet nezvýšil. Zabrání tak častému překreslování, ke kterému by docházelo při psaní textu či pohybu boxu. Nahoře nalezneme funkce pro prezentování, náhled toho jak slajdy budou vypadat při prezentování, poté tlačítka konzole, zobrazení mřížky a tlačítka zpět a vpřed. V pravém horním rohu je pak vynechané místo pro menu, které se objevuje, tehdy když je aktivní textový box.



Obrázek 3: Rozložení úvodní stránky.

Aplikace se snaží být konzistentní co se týče vyskakovacích oken a hlášek, na které uživatel narazí. Okna se objevují ve středu obrazovky, obsahují popsané vstupy a jsou ukončeny dvěma tlačítky (obrázek 6). Hlášky se dělí na tři typy a jsou barevně odlišené – varovné (červená), informativní (modrá) a oznamující úspěch (zelená). Objevují se pak v horní části obrazovky (například obrázek 9). Pro zavírání slouží tlačítka a klávesa ESC. Hlášky zobrazující pouze informaci nebo úspěch, pak mizí po časovém limitu samy.

5.2 Vytvoření slajdu

Pro práci se slajdy uživatel použije část v levém menu nadepsanou „SLAJD“. Zde vybere požadovanou akci, v tomto případě „Nový“, který umístí nově vytvořený slajd do kontejneru, ale i do náhledu vpravo. Nově vytvořený slajd je zmenšený na 50%, ale toto zmenšení lze pak, pomocí tlačítek „Zvětšit“ a „Zmenšit“, měnit. Velikost má danou rozměry, které získal z rozlišení uživatele. Ta je pro celou prezentaci stejná. Pro změnu rozměrů je nutné změnit rozlišení celé prezentace v části „PREZENTACE“ položkou „Rozlišení“. Vytvořenému slajdu lze také měnit i barva pomocí „Pozadí“, které otevře okno pro úpravu barev. Dalším způsobem, jak může uživatel vytvořit slajd, je zkopírování již existujícího slajdu a jeho následné vložení. Tohoto úkonu se dá jednoduše docílit pomocí položek „Kopírovat“ a „Vložit“ v části „SLAJD“.

5.3 Změna pořadí slajdů

Změna pořadí slajdů se provádí pomocí pravého panelu, který obsahuje náhledy jednotlivých slajdů. Uživatel zde může mezi slajdy přepínat, ale také pomocí červených šipek slajdy posouvat v pořadí nahoru či dolů.

5.4 Zpět, vpřed

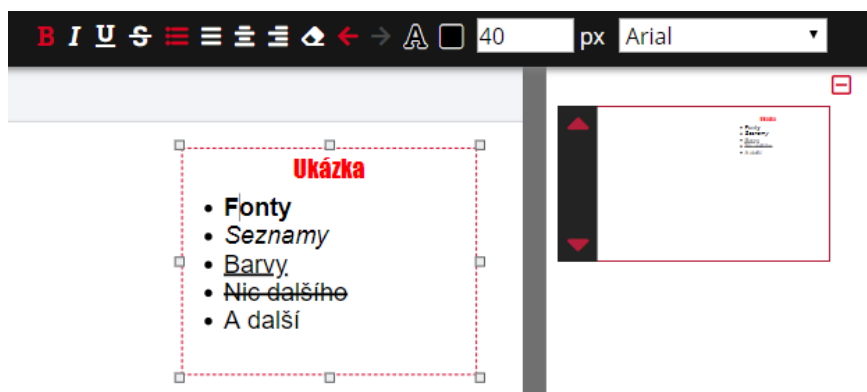
Aplikace nabízí dvě možnosti tlačítek zpět a vpřed pro jednodušší používání. První možnost se nachází v horním menu vlevo a vztahuje se k akcím boxů, slajdů a samotné prezentace. Druhé zpět a vpřed, které jsou umístěny vpravo nahoře v nabídce textových boxů, slouží pro změny v rámci daného textového boxu. Díky tomu uživatel nemusí procházet změnami celé prezentace - například kvůli rozhodnutí vrátit styl textu. Klasické klávesové zkratky CTRL+Z a CTRL+Y se vztahují pouze k textovým boxům, aby se předešlo zmatení uživatele, způsobené nejistotou, ke které akci se zkratka vztahuje.

5.5 Vytvoření prvku

Obsah slajdů je tvořen tzv. prvky. Ty jsou rozlišeny na tři typy - textový box, obrázek a video. Toto rozdělení je dáno podle toho co obsahují. Uživatel pracuje se všemi stejně a to pomocí položek ze sekce „PRVEK“. Prvkům je možné,

podobně jako slajdu, změnit barvu, průhlednost a je navíc možné přidat i zaoblení. Další možností je přidání rámečku. Rovněž je zde možné změnit vrstvu a to pomocí tlačítek „Zvýšit vrstvu“ a „Snížit vrstvu“. Tím může uživatel umístit dříve vložený prvek před nově vložený. Nový prvek lze vytvořit položkami v sekci „PRVEK“ pomocí položek:

- **Textový box** – vloží do aktuálního slajdu nový a prázdný textový box. Ten, na rozdíl od jiných prvků, má vlastní nabídku v pravém horním rohu, obsahující základní možnosti úpravy písma – změnu na tučné, šikmé, podtržené či přeškrtnuté písmo. Dále možnost vytvořit seznam (s odrážkami nebo bez, díky čemuž si pak uživatel může určit číslování), změnit zarovnání, barvu, velikost písma, font a nebo vymazat formátování. Pro pohodlné stylování je doporučeno nejdříve text napsat a až poté ostylet.



Obrázek 4: Příklad textového boxu.

- **Obrázek** – při vkládání obrázku je zobrazeno okno 5, které se liší dle toho, zda je již prezentace uložena nebo ne. V prvním případě má uživatel možnost na server nahrát svůj vlastní obrázek (toto nahrávání je chráněno heslem) nebo vložit obrázek jen pomocí adresy. Uživatel neuložené prezentace může obrázek vložit jen z URL adresy.
- **Video** - vložením videa se uživateli otevře okno podobně jako u obrázku. Nabízí však stejné možnosti a to neohledně na uložení – vložení videa z YouTube²⁹ a nebo vložení adresy videa. Podporované formáty jsou mp4 a ogg.

Dalším způsobem vytvoření je, podobně jako při slajdu, kopírování daného boxu pomocí „Kopírovat“ a následného „Vložit“.

Všechny vytvořené prvky se objeví ve slajdu umístěném v kontejneru a také v náhledu vpravo. Se souřadnicemi a velikostí prvku lze pak intuitivně přímo pracovat posouváním myši, či pomocí vstupů v dolní části levého menu. Pro jednodušší a přesnější práci je možné na slajd umístit mřížku pomocí ikony „mnoha čtverečků“ v levém horním menu.

²⁹<https://www.youtube.com/>

Uložené obrázky

github.jpg

Nahrát obrázek

Pro nahrání/smazání obrázku je vyžadováno heslo

.....

Upload

URL adresa

Maximálně 250 znaků (gif | png | jpg | jpeg | webp | bmp)

Adresa obrázku

Vložit uložený Zavřít

Obrázek 5: Příklad okna správce obrázků pro uloženou prezentaci.

5.6 Prezentování

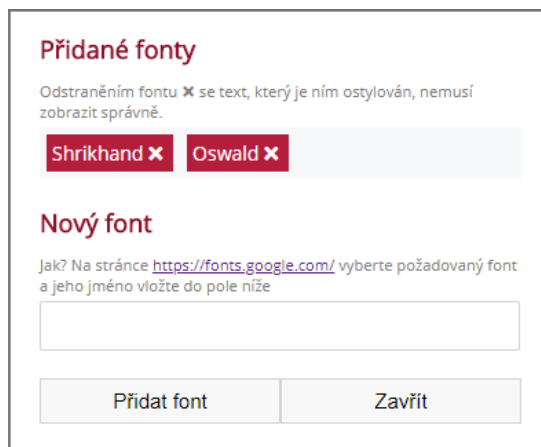
Prezentování se provádí pomocí trojúhelníkové ikony nacházejícího se vlevo nahoře, vedle nápisu „SLIDLY“. Spuštěním tohoto tlačítka se prezentace přepne do režimu celé obrazovky a je zobrazen aktuální slajd. Mezi slajdy pak uživatel může přepínat pomocí šipek na klávesnici nebo myši. Po dosažení posledního slajdu se objeví černý snímek a po dalším kliknutí se obrazovka přepne zpátky do editoru. Ukončit prezentaci lze kdykoliv pomocí klávesy ESC. Pro náhled, jak budou jednotlivé slajdy při prezentování vypadat, je možnost přepnout se tlačítkem „oka“, vedle tlačítka pro prezentování na verzi, která nebude obsahovat rámečky boxů sloužící pro změnu velikosti a určení polohy prvku.

5.7 Správa fontů

Prezentace obsahuje množinu základních fontů. Tuto množinu lze doplnit o fonty nové, které nabízí Google³⁰. Tato funkce je přístupná pomocí možnosti „Správa fontů“ v sekci „PREZENTACE“, která zobrazí okno 6 pro přidání a odstraňování samotných fontů. Takto přidané fonty lze pak používat napříč celou prezentací. Uživatel je může nalézt v menu textového boxu mezi ostatními nabízenými fonty

³⁰<https://fonts.google.com/>

i v PDF editoru. Díky tomu je uživatel zbaven problémů s hledáním fontu mezi ostatními, často nepotřebnými fonty, které by se mu ani nemuseli zobrazit. Fonty je potřeba přidat pouze jednou, při dalším otevření prezentace se automaticky načtou.



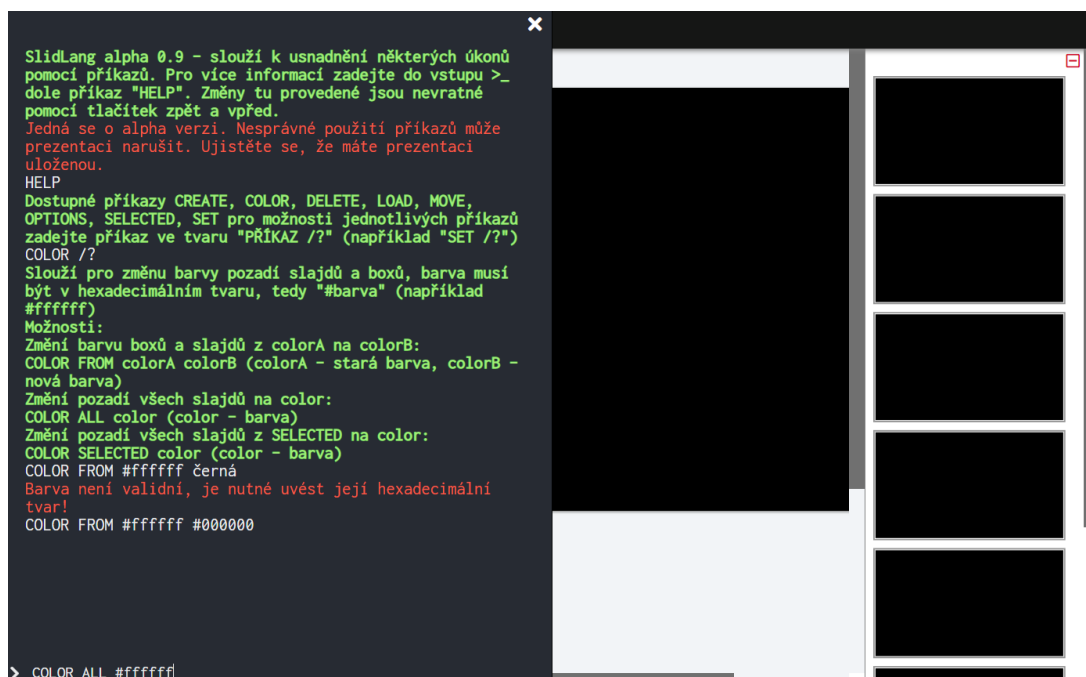
Obrázek 6: Příklad okna správce fontů.

5.8 Pokročilé možnosti

Aplikace, mimo základní možnosti, nabízí také volby pro pokročilé uživatele. Je ji však možné používat i bez nich. Tyto funkce jsou přístupné pomocí ikony znaménka „větší než“ a „podtržítka“ umístěné v levém horním menu. Tato konzole by měla být intuitivní pro uživatele, kteří pracovali s nějakým příkazovým řádkem. Ovládá se pomocí příkazů, které jsou case sensitive³¹. Ty zjednodušují práci s prezentací - umožní zpracovávat více slajdů najednou, díky čemu je možné například změnit jednu konkrétní barvu ve všech boxech a slajdech na jinou (viz obrázek 7) nebo provést hromadné odstranění. Více informací nabízí konzole samotná po zapsání příkazu HELP.

Důležitou funkcí je možnost vytvářet šablony. Uživatel si zde může nastavit objekt „OPTIONS“, který má vlastnosti představující úvodní slajd, slajd s titulkem, běžný slajd a závěrečný slajd. Těmito vlastnostem může dát čísla (čísluje se od 0). Po nastavení pak může pomocí CREATE vytvořit šablonu. Ta je ve formě JSON a uživatel si ji zkopíruje. Pro použití šablony pak stačí do konzole napsat příkaz LOAD, mezeru, za ní vložit zkopírovanou šablonu a provést danou akci. Po správném načtení se objeví LOADED a uživatel pak může vytvářet slajdy vypadající jako ze zdrojové prezentace. Například příkazem LOAD common se do prezentace vloží nový běžný slajd. Příkazy OPTIONS a LOAD by šly využít i na zkopírování prezentace tím, že by jsme všechny její slajdy dali do objektu „OPTIONS“. Takto by šlo však zkopírovat jen prezentaci o maximálně 4 slajdech.

³¹Citlivost na velikost písma, tedy např. „b“ a „B“ není to samé.



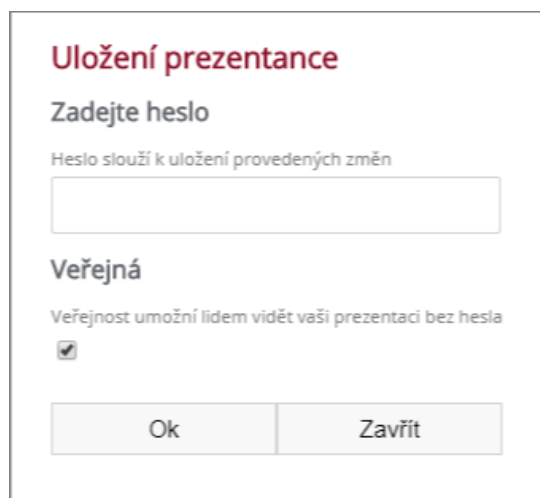
Obrázek 7: Příklad použití konzole.

Důvod pro neumožnění zkopírovat větší prezentaci je snaha zamezit cizímu uživateli zneužívat kopírování kvůli krádeži. Uživatel, jemuž prezentace patří, si ji může zálohovat pomocí tlačítka „Exportovat“ (viz [5.11](#)).

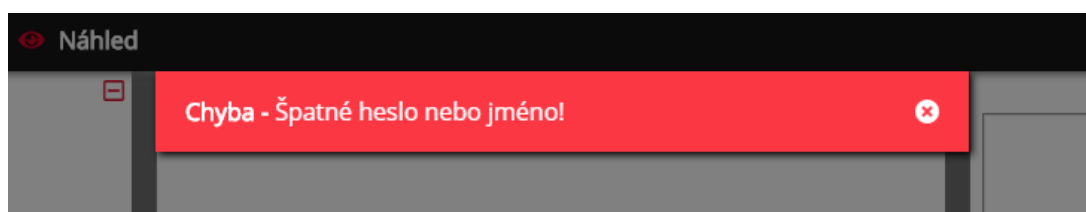
Po ukládání prezentace bez registrace je konzole další z věcí, které nenabízí žádná z mnou nalezených existujících řešení. A ačkoliv se nachází v alpha verzi, již nyní je velmi praktická.

5.9 Uložení

Možnosti, popsané výše, může uživatel používat i bez uložení. V případě, že chce k prezentaci znovu přistoupit, vygenerovat ji do PDF nebo nahrávat vlastní obrázky, je nutné prezentaci uložit. K uložení prezentace se používá okno vyvolané pomocí možnosti „Uložit“ nacházejícího se v sekci „PREZENTACE“. Toto okno se liší pro již uloženou a neuloženou prezentaci. Ta uložená [8](#) obsahuje vstup pro heslo a zaškrtnutí políčko určující, zda je prezentace veřejná. Veřejnost prezentace určuje, zda je možné ji číst bez zadání hesla nebo jestli heslo vyžadovat bude. Neuložená prezentace vyžaduje navíc vstup pro název, pomocí něhož je možné k prezentaci v budoucnu přistoupit. Také se zde nachází textové pole pro potvrzení hesla. Jelikož se jedná o službu bez registrace, neudává se e-mail. Tím pádem není možné heslo později změnit. Prezentace je v databázi uchována po dobu 30 dní od jejího posledního uložení, poté je automaticky odstraněna.



Obrázek 8: Okno uložení již uložené prezentace.



Obrázek 9: Pokus o uložení prezentace se špatným heslem.

5.10 Přihlášení

Při každém uložení je nutné zadávat heslo, což může být nepříjemné pro uživatele, kteří jsou zvyklí své práce často ukládat. Důvodem je to, že Slidly je cílen i na méně pokročilé uživatele, například i děti, ti mohou z nepozornosti, zapomenout zavřít aplikaci, čehož může využít cizí útočník a prezentaci jim upravit. Zadávání hesla se však uživatelé můžou vyhnout tlačítkem „Ukládání s heslem“, které otevře okno, do kterého pak stačí jednou zadat heslo. Díky tomu žádné funkce nebudou vyžadovat heslo. Tato možnost funguje do aktualizace stránky nebo jejího opuštění. Když uživatel otevřel neveřejnou prezentaci, je také zapnut mód „Ukládání bez hesla“, pro který platí stejné podmínky.

5.11 Export, import

Uživatelé, například učitelé, kteří potřebují mít prezentace k dispozici například jen jednou za rok, mohou využít možnosti exportovat prezentaci. Prezentace se exportuje do textového formátu. Ten si mohou zkopírovat a v případě potřeby importovat do aplikace. Export je chráněn heslem, díky čemuž nemůže dojít k nechtěnému zkopírování. Pro provedení těchto operací slouží tlačítka „Exportovat“ a „Importovat“ v sekci „PREZENTACE“.

5.12 PDF

Prezentaci je možné převést do souboru PDF pomocí tlačítka „Vygenerovat do PDF“ nacházejícího se v sekci „PREZENTACE“. Po kliknutí je uživatel přemístěn na novou stránku, která se podobá té předchozí. Neobsahuje však levé menu. Kontejner na slajdy, které jsou v náhledovém režimu, je na levé straně. Vpravo se nachází nový panel, který po otevření nabízí možnosti pro vytvoření záhlaví či zápatí samotných slajdů. Je možné mu měnit barvu, velikost písma, font, umístění či průhlednost. Při změně textu je možné použít speciální řetězce, které jsou nahrazeny informacemi. Mezi tyto speciální řetězce patří „*poradi*“, který se změní na aktuální číslo slajdu, „*pocet*“, ten je nahrazen celkovým počtem slajdů a také „*datum*“ nabývající hodnoty aktuálního datumu. Vytvoření PDF je pak uživateli přístupné pomocí možnosti „Vygenerovat PDF“.



Obrázek 10: Příklad řetězců prezentace v PDF.

6 Rozšíření aplikace

Aplikace podobného typu nabízí velkou řadu funkcí zpříjemňujících používání. Tyto funkce jsou často přidávány týmy po řadu několika let. Proto jsem se snažil vytvořit ze Slidly spíše platformu, která by podobný vývoj umožnila. Díky rozdělení aplikace do jednotlivých komponent, je přidání nových možností snadné. Boxy, zajišťující pohyb a ovládání obsahu slajdu, jsem vytvořil tak, aby byly nezávislé na svém obsahu, což nabízí možnost přidat téměř jakoukoliv React komponentu během pár desítek minut. Použitím vlastní komponenty pro vytváření vyskakovacích oken jsem se také zbavil povinnosti vytvářet novou komponentu pro každé základní okno, které by bylo v budoucnu nutné přidat.

Server je také připraven pro budoucí rozšíření. Jednotlivé služby jsou rozděleny do směrovačů, které spolu nesouvisí. Přidáním jednoho řádku je pak možné jednoduše využít směrovač nový. Server nabízí i veřejné API pro případné další aplikace.

Díky jednoduchosti přidat nový prvek se nabízí hned několik možností, které bych rád v budoucnu implementoval nebo zlepšil:

- Přidání nových typů prvků, jako je například audio, tabulka, graf či kreslicí plátno.
- Zlepšit podporu prohlížečů.
- Registrace. Ačkoliv si stále myslím, že možnost vytvářet prezentace rychle a bez přihlášení je důležitá, zdá se mi vhodné vytvořit dobrovolnou registraci umožňující uživateli možnost změnit heslo, spravovat jeho prezentace či poskytnou je jen nějakým uživatelům.
- S předchozím bodem souvisí také možnost víceuživatelského přístupu, tedy možnosti, aby více uživatelů upravovalo danou prezentaci v jeden moment. Tu jsem kvůli chybějící registraci implementovat nemohl.
- Vytvořit responzivní verzi, aby se prezentace daly vytvářet pohodlně i jinde než na počítači.
- Umožnit vkládat URL adresy a používat CSS 3 transformace, které jsem neimplementoval kvůli generování do PDF.
- Přidat animace pro lepší zážitek uživatelů.
- Lokální ukládání, aby uživatel mohl prezentovat i offline.
- Vylepšení konzole tak, aby nabízela více funkcí. Přidat podporu vlastních skriptů nebo možnost získávat data z dostupných API, a tím vytvořit prezentaci, která bude například každý den měnit svůj stav.

Závěr

Výsledkem práce je webová aplikace pro tvorbu prezentací. Aplikace se snaží vyhnout problémům již existujících řešení a přichází s novou koncepcí ukládání prezentací bez registrace. Uložená prezentace je dostupná prostřednictvím uživatelem zadané URL adresy. Většina podobných aplikací je pro běžné uživatele příliš komplikovaná, protože obsahují zbytečné funkce, které většinu času dělají práci matoucí. Moje řešení je nemá, stále však poskytuje stejný komfort ovládání. Nabízí také nové zajímavé funkce, jako je konzole nebo přidávání fontů. Uživatel potřebuje pouze moderní webový prohlížeč a internet. Aplikace je však také připravena pro prezentování bez nich. Umožňuje totiž generovat prezentaci do PDF souboru. Díky těmto vlastnostem aplikace splňuje jednoduchost, přenosnost a vytrvalost, což byly hlavní požadavky.

Díky návrhu a použití React komponent je relativně snadné aplikaci rozšířit. Tyto komponenty, které jsem sám vytvořil, jsou použitelné i pro podobné projekty, jako je grafický editor nebo textový procesor. Rozšíření jsou k dispozici také na straně serveru, který nabízí systém směrovačů a rozhraní REST API umožňující snadné vytváření nových klientských aplikací se stejnými daty.

Svět webových aplikací byl pro mě zcela nový, takže kvůli této práci jsem se toho hodně naučil. Vyzkoušel jsem nové technologie a přístupy. Zlepšil jsem své dovednosti zabezpečení, návrhu i grafiky. Tyto dovednosti budu nadále využívat jak v nových projektech, tak i ve zlepšování tohoto, který plánuji v budoucnu rozšířit.

Conclusions

The result of the thesis is the web application for slideshows creation. The application tries to avoid the problems of existing solutions and comes with new concept of registration-free slideshows storage. Saved slideshow is accessible via user-specified URL address. Most of the other applications are too complicated for common users because they contain unnecessary functions, which make the work confusing for most of the time. My solution does not include such functions, but it still offers the same comfort of control. It also offers new interesting features such as console or adding of fonts. User only needs a modern web browser and the internet connection. However, the application is also ready for presentation without them, because it can generate slideshow into a PDF file. Thanks to these features the application meets simplicity, portability and persistence, which were the main requirements.

Thanks to the design and use of React components, it is relatively easy to expand the application. These components, which I have created by myself, are also usable for similar projects such as graphics editor or word processor. Extensions are also available for a server-side that offers a router system and REST API to provide easy creation of new client applications with the same data.

The world of web applications was new to me, so thanks to this thesis, I have learned a lot. I tried new technologies and approaches. I have improved my security, design and graphic skills. I will continue to use these skills in new projects and in improving this one too, which I plan to expand in the future.

A Instalace a spuštění serveru

A.1 Požadavky:

Pro spuštění serveru lokálně je nutné mít:

- Nainstalovaný Node.js (k vývoji byla použita verze v8.11.3, dostupná na adrese <https://nodejs.org/en/download/>)
- Nainstalovaný správce balíčků npm, který je součástí instalace Node.js
- Neobsazený port 8080

A.2 Spuštění serveru:

Pro spuštění serveru proveďte v adresáři server na vašem disku příkazy v příkazovém řádku:

1. `npm install` (stažení požadovaných závislostí)
2. `npm start` (spuštění serveru)

Po správně provedeném startu server běží na adrese <http://localhost:8080/> a do příkazového řádku by mělo být vypsáno:

```
Posloucham na portu 8080!  
Database uspesne pripojena!
```

B Instalace a spuštění aplikace:

B.1 Požadavky

Pro spuštění aplikace je nutné mít:

- Nainstalovaný správce balíčků npm viz. [A.1](#)
- Neobsazený port 3000

B.2 Spuštění aplikace:

Pro spuštění aplikace proveďte v adresáři aplikace příkazy v příkazovém řádku:

1. `npm install` (stažení požadovaných závislostí)
2. `npm start` (spuštění aplikace)

Po správně provedeném startu aplikace by mělo být do příkazového řádku vypsáno `Compiled successfully!`, informace, kde je aplikace dostupná a varování, že běží ve vývojovém módu a není tedy optimalizovaná.

C Testování

Pro účely testování při tvorbě posudků práce je možné využít lokální instalace (viz přílohy A a B) nebo také veřejně dostupné verze zprovozněné na studentském serveru:

- server - <https://tux.inf.upol.cz/nodejs/>
- aplikace - <https://tux.inf.upol.cz/~valipe00/>

Aplikace byla vyvíjena a testována v prohlížeči Google Chrome a to ve verzi 67.0.3396.99 na 64bitovém Windows 10 Home. Dále proběhly rychlé testy v Microsoft Edge 42.17134.1.0 a Firefox 61.0.1, u nich a ostatních však není možné zaručit plnou funkčnost.

D Obsah přiloženého CD/DVD

Přiložené CD/DVD obsahuje aplikaci a digitální verzi bakalářské práce.

doc/

Text práce ve formátu PDF, včetně všech příloh a všech souborů potřebných pro bezproblémové vygenerování PDF dokumentu textu.

src/

Zdrojové kódy webové aplikace SLIDLY její serverové části i klientské části.

readme.txt

Instrukce pro instalaci, spuštění serveru a webové aplikace, včetně požadavků pro jejich bezproblémový lokální provoz. Dále se zde nachází webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce.

Literatura

- [1] Microsoft PowerPoint 2016 – software pro prezentace [online]. [cit. 2018-07-26]. Dostupné z: <https://products.office.com/cs-cz/powerpoint>
- [2] Prezi [online]. [cit. 2018-07-26]. Dostupné z: <https://prezi.com/>
- [3] Prezentace Google – zdarma vytvářejte a upravujte prezentace online. [online]. [cit. 2018-07-26]. Dostupné z: <https://www.google.cz/intl/cs/slides/about/>
- [4] About JavaScript - JavaScript | MDN [online]. [cit. 2018-07-26]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
- [5] Classes - JavaScript | MDN [online]. [cit. 2018-07-26]. Dostupné z: <https://developer.mozilla.org/cs/docs/Web/JavaScript/Reference/Classes>
- [6] AJAX Introduction [online]. [cit. 2018-07-26]. Dostupné z: https://www.w3schools.com/xml/ajax_intro.asp
- [7] HTML5 [online]. [cit. 2018-07-26]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
- [8] CSS3 [online]. [cit. 2018-07-26]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>
- [9] About [online]. [cit. 2018-07-26]. Dostupné z: <https://nodejs.org/en/about/>
- [10] What is the Event Loop? [online]. [cit. 2018-07-26]. Dostupné z: <https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/>
- [11] Express - Node.js web application framework [online]. [cit. 2018-07-26]. Dostupné z: <https://expressjs.com/>
- [12] Using Express middleware [online]. [cit. 2018-07-26]. Dostupné z: <https://expressjs.com/en/guide/using-middleware.html>
- [13] Simple express middleware for uploading files [online]. [cit. 2018-07-26]. Dostupné z: <https://www.npmjs.com/package/express-fileupload>
- [14] React - A JavaScript library for building user interfaces [online]. [cit. 2018-07-26]. Dostupné z: <https://reactjs.org/>
- [15] Draft.js - Rich Text Editor Framework for React [online]. [cit. 2018-07-26]. Dostupné z: <https://draftjs.org/>
- [16] Immutable collections for JavaScript [online]. [cit. 2018-07-26]. Dostupné z: <https://facebook.github.io/immutable-js/>
- [17] What Is MongoDB? [online]. [cit. 2018-07-26]. Dostupné z: <https://www.mongodb.com/what-is-mongodb>

- [18] Mongoose [online]. [cit. 2018-07-26]. Dostupné z: <http://mongoosejs.com/>
- [19] HERTZEN, Niklas von. Html2canvas: Screenshots with JavaScript. GitHub [online]. [cit. 2018-07-26]. Dostupné z: <https://github.com/niklasvh/html2canvas>
- [20] HALL, James. JsPDF: Client-side JavaScript PDF generation for everyone. GitHub [online]. [cit. 2018-07-26]. Dostupné z: <https://github.com/MrRio/jsPDF>
- [21] TYPEKIT. Web Font Loader [online]. [cit. 2018-07-26]. Dostupné z: <https://github.com/typekit/webfontloader>
- [22] Font Awesome [online]. [cit. 2018-07-26]. Dostupné z: <https://fontawesome.com/>
- [23] File Structure - React [online]. [cit. 2018-07-26]. Dostupné z: <https://reactjs.org/docs/faq-structure.html>
- [24] bcrypt for NodeJs, npm [online]. [cit. 2018-07-26]. Dostupné z: <https://www.npmjs.com/package/bcrypt>
- [25] MALÝ, Martin. REST: architektura pro webové API [online]. 2009-08-03. [cit. 2018-07-30]. Dostupné z: <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>
- [26] KEMP, John. Security on the Web [online]. 2011-02-04. [cit. 2018-07-31]. Dostupné z: <https://www.w3.org/2001/tag/2011/02/security-web.html>