

Gymnázium, Praha 6, Arabská 14

Obor programování



TEAMOVÝ ROČNÍKOVÝ PROJEKT

Ondřej Czadera, Vojtěch Šletr, Vojtěch Vojtěch

Čejiny.cz

Rádi bychom na tomto místě poděkovali panu profesorovi Mgr. Janu Lánovi, který byl vedoucím naší ročníkové práce. Děkujeme mu za pomoc a vstřícnost během celého psaní a za ochotu korigovat naše nápady i chyby, především v rámci hostingu naší ročníkové práce.

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

Ondřej Czadera

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Kladně dne

Vojtěch Šletr

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Kladně dne

Vojtěch Vojtěch

Název práce: Čejiny.cz

Autoři: Ondřej Czadera, Vojtěch Šletr, Vojtěch Vojtěch

Anotace: Cílem této práce je vytvořit webový portál pro učení se českých dějin. Tento portál bude obsahovat lekce na témata českých dějin a evropských událostí ovlivňujících české dějiny do konce druhé světové války. Uživatel se bude přihlašovat ke svému účtu a po přečtení lekce může uživatel absolvovat testy s hodnocením úspěšnosti k danému tématu a testy dle libosti opakovat, aby se zlepšoval. K tématům budou dostupné zdroje pro pokračování ve vzdělání, které k tématu doporučujeme. Formáty testu budou a-b-c, spojování pojmů a seřazení dle časové posloupnosti.

Project name: Čejiny.cz

Authors: Ondřej Czadera, Vojtěch Šletr, Vojtěch Vojtěch

Annotation: The goal of this project is to create a web portal for learning about Czech history. This portal will contain lectures about events from Czech history and European events, that in some way affected Czech history. The lectures will cover the period from the dark ages until the end of the second world war. The user will be able to login into his account and after reading through the lecture, he will be able to take a test about the topic of the lecture. He will afterwards get his results and, if unsatisfied with his results, will be able to take the test again as many times as he wishes. We will also provide some sources with the lectures, from which the user could further deepen his knowledge about the topic. The format of the tests will be a-b-c, name connection and sorting by the time of the events.

Zadání

Zadáním tohoto projektu bylo vytvořit webový portál a vytvořit mu obsah. Na tomto portálu bude mít uživatel možnost prohlížet si jednotlivé lekce a po přihlášení bude mít uživatel možnost napsat test, kterým své znalosti z lekce může otestovat. Formát testů bude a-b-c, spojování pojmů a řazení na časové ose.

Obsah

1.	Přehled projektu	1
1.1.	Úvod	1
1.2.	Volba použité technologie	1
1.3.	Nastavení projektu a hosting	2
1.4.	Vzhled stránek	2
1.5.	Přihlášení uživatele	3
2.	Admin panel	4
3.	Testy	5
3.1.	Vytváření testů	5
3.2.	Oprava testových otázek	6
3.3.	Vypsání testu pro uživatele	7
3.4.	Průběh testu	8
4.	Lekce	10
4.1.	Vytváření lekcí	10
4.2.	Vypsání lekcí	12
5.	Závěr	13
6.	Bibliografie	14
7.	Seznam obrázků	14
8.	Přílohy	15
8.1.	Seznam literatury použité na tvorbu lekcí	15

1. Přehled projektu

1.1. Úvod

Lekce tvoří společně s testy většinu obsahu našich stránek. Všechny jsme je vytvářeli sami a k datu odevzdání tohoto projektu je dokončeno 10 kompletních lekcí a ke každé je zhotoven i test. Celkem mají lekce dohromady délku 35 825 znaků, což je přibližně rozsah ročníkové práce z druhého ročníku. Tyto lekce pokrývají víceméně celé období vymezené zadáním, a to období od počátků lidského osídlení v Čechách až po události, které vedou k druhé světové válce.

1.2. Volba použité technologie

Na samotném začátku našeho projektu byla první otázkou technologie, kterou využijeme k vytvoření stránky čejiny.cz. Nikdo z nás neměl žádné zkušenosti s vytvářením dynamických HTML stránek, a proto jsme se snažili najít pro nás co nejlehčí způsob, jak na to. Nakonec jsme zvolili JavaServer Pages (JSP). „JavaServer Pages je technologie pro vývoj hlavně dynamických HTML stránek založená na jazyce Java. Vyvinuli jí v Sun, ale pak ji koupila firma Oracle.“ (Wikimedia Foundation, 2020) Využití JSP byla pro naše účely nejlepší možnost, protože všichni z našeho teamu programujeme v jazyce Java a díky JSP ji můžeme využít k programování dynamických webových stránek.

Java se pomocí JSP v HTML souborech používá tak, že se pomocí oddělovačů¹ oddělí Java a HTML kód. Běžný uživatel nepozná, že se jedná o JSP stránky. Podívá-li se do zdrojového kódu, neuvidí žádnou Javu, protože se ve zdrojovém kódu neukazuje, pouze HTML.

Další věc, kterou jsme museli zvolit, byl databázový systém, který se bude využívat k zápisu testů, informací o lekcích, výsledků testů a také jako úložiště přihlašovacích údajů admin uživatelů. Po chvíli bádání na internetu jsme se shodli, že využijeme PostgreSQL. Program využívaný ke správě těchto databází se nazývá pgAdmin. Díky tomuto programu si můžete spustit vlastní PostgreSQL databázi na „localhostu“², a také se připojit k databázi hostované někým jiným, máte-li přihlašovací údaje.

¹ <% Java kód %>

² Místní počítač, na kterém pracujete

1.3. Nastavení projektu a hosting

Po zvolení technologie, kterou budeme používat, jsme museli zvolit hosting, na kterém tento projekt spustíme. Rozhodli jsme se, že k tomuto účelu použijeme Google Cloud Platform, který se chlubí tím, že vše, co potřebujeme, tj. JSP stránky a PostgreSQL databázi, dokáže spravovat ze své konzole. Tato volba se ale ukázala jako špatná, protože jsme se nebyli schopni připojit k databázi.

Nakonec jsme tedy přešli na Apache Tomcat, což je „open source webový server a servlet kontejner.“ (Wikimedia Foundation, 2019) Jako hosting pro Tomcat nám bylo doporučeno, panem profesorem Lánou Heroku, které poskytuje hosting jazyků jako je právě Java, Node.js, Scala, Clojure, Python, PHP, Ruby a Go. (Wikimedia Foundation, 2020)

Hosting na Heroku si můžete buď zaplatit, anebo se pokusit vyjít s jejich „free“³ balíčkem, který je naštěstí pro náš projekt naprosto dostačující. Jediné dvě nevýhody „free“ balíčku na Heroku jsou, že naše doména, na které se „čejiny“ právě nacházejí, není <https://cejiny.cz>, ale <https://cejiny.herokuapp.com>, a že tato doména nefunguje stále, takže když uživatel zadá naši url do prohlížeče, tak musí chvíli vyčkat, než se na heroku serverech pustí instance „čejin“ a teprve potom se mu stránky zobrazí.

Jako vývojové prostředí jsme využívali IntelliJ IDEA od JetBrains, jelikož má nativní podporu Apache Tomcat a PostgreSQL. Se sdílením našeho projektu mezi sebou nám pomohl GitHub, kde jeden z nás vytvořil repozitář, kam jsme poté začali nahrávat projektové soubory.

1.4. Vzhled stránek

Vzhled stránek, tak jak ho vidí běžný uživatel, byl vytvořen pomocí CSS frameworku W3.CSS, který byl podle autorů vytvořen jako alternativa k Bootstrapu. W3.CSS je menší, rychlejší, lehčí na práci a nezávislá na jQuery či jakékoli další JavaScript knihovně. (Refsnes Data, 2020) Pomocí tohoto frameworku jsme vytvořili základní vzhled stránky (menu, rozvržení), který jsme poté následovali ve všech dalších částech stránky. Pro získání ikonek v menu a také v admin části jsme využili open source projektu Font Awesome, který nabízí něco přes 1500 „free“ ikon⁴.

³ zadarmo

⁴ s placenou verzí pomalu 8000 ikon

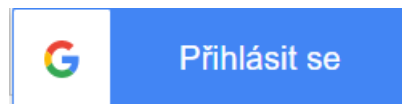
Hlavní logo a „favikona“⁵ byla vytvořena v programu Adobe Photoshop. „Favikona“ je Č, jako čejiny, a uprostřed tohoto Č je mapa ČR s „trikolórou“ české vlajky, na obrázek této mapy se vztahuje Pixabay licence a není nutné uvádět zdroj.

Základní rozměry stránky jsou nastaveny v souboru „sitesize.css“, na který se odkazuje soubor „menu.jsp“, který pomocí JSP funkce „include“ přidáváme do každé další stránky, kde chceme vidět základní menu, takže cokoliv, co upravíme v tomto souboru, se okamžitě projeví na všech dalších, normálnímu uživateli přístupných, stránkách.

Vzhled menu se v průběhu, kdy uživatel pracuje se stránkou, může změnit. Jakmile si někdo otevře naši stránku, zobrazí se na hlavním menu pouze dvě možnosti, Lekce a Účet, aby uživatel viděl i záložku testy, tak se musí přihlásit ke svému google účtu v záložce Účet. Testy nepřihlášený uživatel nevidí, kvůli ukládání výsledků těchto testů, které se ukládají pod id přihlášeného uživatele. Poté co se uživatel přihlásí, odemkne se tedy záložka testy, kam již může normálně přistupovat a také se uprostřed zobrazí jméno uživatele⁶. V této chvíli se již dá pohybovat všude, kam má normální uživatel přístup, tj. lekce podle kategorií, seznam všech lekcí, testy podle kategorií, seznam všech testů a záložka Účet.

1.5. Přihlášení uživatele

K přihlášení se do „čejin“ musí mít návštěvník založený Google účet, pomocí kterého se přihlásí přes záložku Účet v hlavním menu.



Obrázek 1 - Google tlačítko k přihlášení

V této záložce je tlačítko Přihlásit se, které otevře novou kartu s přihlášením do vašeho Google účtu. Ihned po přihlášení uživatele si JavaScript⁷ zjistí jméno a id přihlašujícího se návštěvníka, poté skryje tlačítko Přihlásit se a odkryje záložku Testy, tlačítko Odhlásit se a uprostřed menu jméno uživatele. Dál si uloží id přihlášeného uživatele, díky čemuž poté můžeme ukládat výsledky testů každému zvlášť.

Přihlášení přes Google účet je možné díky použití Google Cloud Platform, na této platformě měl na začátku být celý projekt. V Google Cloud je nutné vytvořit nový projekt a nastavit domény, ze

⁵ Ikonka, která se ukazuje v horní liště u spuštěné stránky

⁶ Pomůcka: pokud se jméno ukazuje, uživatel ví, že je přihlášen.

⁷ Soubor: googleintegration.js

kterých se bude dát přihlašovat. Google má přihlašování přes jejich účty ošetřené z obou stran, Google Cloud vygeneruje totiž token, který musí být přímo v HTML kódu stránky, ze které se chcete přihlásit a přímo v Cloudu musí být nastavena doména, ze které tyto požadavky má Google řešit. U nás jsou to tedy dvě domény: „localhost“ a <https://cejiny.herokuapp.com>. Localhost je zde nastavený primárně, aby přihlášení fungovalo, i když se zkusíme přihlašovat z lokálního serveru, a ne přímo z nastavené internetové domény přístupné pro všechny. Nový kód potřebujeme také otestovat, než ho hodíme na oficiální stránku. Toto všechno jsme nastavili s pomocí [tutoriálu](#) přímo od Googlu, kde přesně popisují, jak nastavit projekt v Google Cloud, jak vygenerovat přihlašovací tlačítko a dále nakládání s daty uživatele.

2. Admin panel

V menu není žádný odkaz na přihlášení jako admin uživatel, chce-li se tedy návštěvník přihlásit jako admin, musí znát přímý odkaz⁸ na přihlášení do admin panelu. Soubor „login.jsp“ s pomocí pár dalších java souborů⁹ obsluhuje přihlášení do admin panelu.

Servlety, které přihlašování obsluhují, si vezmou přihlašovací údaje zadané uživatelem ze souboru login.jsp, připojí se do databáze a zjistí, jestli jsou zadané údaje správné a uživatel tedy může být přihlášen. Když jsou zadané údaje správné, tak je uživatel úspěšně přihlášen a přesměrován na stránku „home.jsp“. Jestliže však údaje správné nejsou, napíše se chybová hláška o nesprávnosti údajů a uživatel může pokračovat zpět na hlavní stránku, nebo se pokusit znovu přihlásit. Celý „login.jsp“ soubor je samozřejmě kontrolován ještě pomocí jQuery¹⁰, které hlídá, aby uživatel vždy zadal nějakou přezdívkou a heslo.

Pokud by nepřihlášený uživatel znal přímou adresu na soubor pod admin složkou, tak ho autentikační filtr¹¹ přesměruje znovu na login.jsp. Tato chyba, která by bez tohoto filtru nastávala je tedy zničena.

⁸ <https://cejiny.herokuapp.com/admin/login.jsp>

⁹ Soubory: User.java, UserDao.java, UserLoginServlet.java, k vytvoření těchto souborů jsme použili tento [tutoriál](#) (Minh, Codejava.net, 2019)

¹⁰ Soubor: loginform.js

¹¹ Soubor: AdminAuthenticationFilter.java - tento soubor jako jediný je stažený přímo z tohoto [odkazu](#) (Minh, Codejava.net, 2019) a trochu poupraven k našim potřebám

Po úspěšném přihlášení je uživatel přesměrován na stránku „home.jsp“, kde je rozcestník možností přihlášeného uživatele. První možností je se zase odhlásit. Tlačítko „Logout“ odhlásí uživatele a přesměruje ho na stránku „login.jsp“. Dále je zde možnost „Přidat test“, která „admina“ přesměruje na výpis lekcí bez vytvořených otázek. Tlačítko „Lekce“ přesměruje uživatele na stránku „lekce.jsp“, kde je výpis všech známých informací o již vytvořených lekcích a možnost vytvořit novou lekci. Ve výpisu lekcí jsou informace jako ID lekce, název lekce¹², cesta k lekci¹³, kategorie lekce a jestli je již vytvořený test.

ID	NAME	PATH	CATEGORY	TEST
4	Křest v Řezně	/lekce/krest_v_rezne.jsp	stredovek	ano
1	Úplné počátky	/lekce/uplne_pocatky.jsp	pravek	ano

Obrázek 2 - tabulka lekcí v admin panelu

Poslední možnost na stránce „home.jsp“ je „Editor Otázek“, ve kterém se dají editovat již vytvořené otázky k testům.

3. Testy

Tato část dokumentace se bude věnovat jednomu z hlavních témat naší aplikace, a to testům. Jejich vytváření, zápisu, přenosu k uživateli a jejich zpracování s následným vyhodnocením. Všechn kód potřebný k této části jsem převzal pouze z dokumentace k JSP a ze Stackoverflow.com, což nejsou zdroje vyžadované k citaci

3.1. Vytváření testů

Vytváření testů je možné pouze z účtu administrátora. Ani přihlášený uživatel na hlavní straně nemá k vytváření přístup. V menu pro administrátory se po úspěšném přihlášení nachází přímo odkaz pro vytváření.

Prvotně se spustí soubor „addNewTest.jsp“, který vytvoří spojení s databází a vypíše list lekcí, u kterých nebyly dosud vytvořeny všechny otázky v celkovém počtu 10 otázek. U každé vypisované lekce pak projde ještě jednou databází a prověří, zda již nebyly přidány některé otázky dříve. Tím předchází vzniku „mrtvých“ otázek, které by mohly vznikat při opuštění stránky během tvorby otázek nebo pádu samotné stránky. Bez tohoto ošetření by bylo nutné pokaždé vytvořit všech 10

¹² V té samé chvíli je to i odkaz, na který když uživatel klikne, tak ho to přesměruje na vybranou lekci.

¹³ Ve tvaru: /lekce/nazev_lekce.jsp

otázek během jedné návštěvy. Pokud by k tomu nedošlo, bylo by potřeba postupovat znovu od začátku, avšak v databázi by zůstaly i předchozí, což by vytvářelo neshody v pozdějším načítání samotného testu uživatelem. Po tomto průchodu tedy program dosadí množství již existujících otázek a administrátor může pokračovat ve vytváření bez dalších budoucích omezení.

Po kliknutí na zvolenou lekci, pro kterou chce administrátor otázky vytvářet se spustí funkce jménem „loadTestInput“ v JavaScript souboru jménem „addTest.js“ s atributy jméno lekce, id lekce a počet již existujících otázek. Funkce ze stejné stránky odstraní vypsané lekce bez testu a nahradí je volbou druhu otázky.

Po zvolení druhu otázky se spustí funkce „addingForm“ ve stejném souboru JavaScript, a ta podle druhu vypíše potřebná vstupová pole. Když administrátor otázku sepíše do polí vstupu, jsou k dispozici dvě tlačítka „Uložit Otázku“, které je již aktivní od zvolení typu otázky, a pak „Přejít na další otázku“, které je nepřístupné, dokud není otázka uložena prvním tlačítkem.

První tlačítko všechny zadané informace odešle do souboru „saveQuestion.jsp“. Tento soubor se rychle otevře jako nová záložka prohlížeče, všechna data přečte a uloží do databáze společně s id lekce, ke které patří. Poté se záložka zase uzavře. Tím se zpřístupní druhé tlačítko, které opět načte volbu typu otázky. Ačkoliv není otevírání nové záložky zcela optimálním řešením, vyšlo ze zkoušených možností nejlépe, protože dovoluje zůstat na samotné stránce se zadáváním otázek a nevyžaduje další „prokliky“, jako kdyby se ukládání otázky provádělo na stejné záložce.

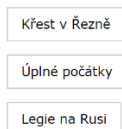
Když po uložení poslední neboli 10. otázky klikne administrátor na druhé tlačítko, načte se finální obrazovka odkazující do admin menu a v databázi se přepíše hodnota lekce tak, že již se nadále nedají k ní vytvářet otázky, a naopak z ní může obyčejný přihlášený uživatel splnit test.

3.2. Oprava testových otázek

Pokud by při tvorbě vznikla otázka, obsahující gramatickou chybu, nebo by jinak nevyhovovala představě, má administrátor po dokončení všech otázek v daném tématu možnost úpravy dané otázky. V menu administrátora je přímo odkaz pro úpravu. Nejdříve uživatel vybere téma, pod které otázka spadá a pak samotnou otázku. Na stránce „/QuestionEdit.jsp“ se načtou pole vstupu s hodnotami uloženými v databázi, která jsou nyní upravovatelná. Také je možnost změnit kategorii otázky, zde se ale již nepřenáší data z původní otázky. Když je administrátor s úpravou

spokojen, stiskem tlačítka „Změnit otázku“ proběhne aktualizování záznamu v databázi a uživatel se může vrátit zpět do menu.

Výpis testů s možností editace



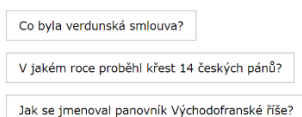
Křest v Řezně

Úplné počátky

Legie na Rusi

Obrázek 3 - Volba editace otázky pod daným tématem

Výpis otázek s možností editace na téma: Křest v Řezně



Co byla verdunská smlouva?

V jakém roce proběhl křest 14 českých pánů?

Jak se jmenoval panovník Východofranské říše?

Obrázek 4 - Volba samotné otázky pro editaci

3.3. Vypsání testu pro uživatele

Tato funkce je povolena pouze přihlášeným uživatelům, a to z důvodu ukládání výsledků, které jsou navázány na uživatelské id Google účtu. K samotným testům se může uživatel dostat v hlavičce našich stránek pod záložkou testy, kde si může zvolit daný test buď podle kategorie ve které se dané téma nachází, anebo v celkovém výpisu.

Ať už zvolí jakoukoliv možnost, je přesměrován na stránku, kde jsou vypsány lekce, u kterých bylo vytvořeno všech 10 otázek, v případě kategorií je nejdříve přesměrován na stránku s danými kategoriemi a teprve poté dochází k výpisu.

Při výpisu testů poté dochází k procházení databáze pro nalezení předchozích výsledků a vpravo na stejném řádku je pak uvedeno buď maximální dosažené skóre v procentech, anebo, pokud ještě uživatel test nikdy nevyplnil, oznámení, že jej dosud nevyplnil. Pokud by nebyly ve výpisu testů, ať pro určitou kategorii, nebo pro všechny testy, žádné dostupné testy, je vypsáno oznámení informující uživatele, že zatím žádné testy neexistují, aby nedocházelo ke scénáři prázdné stránky a domnění, že se stránka stále načítá.

Výpis všech dostupných testů

Křest v Řezně	Vaše nejvyšší hodnocení: 100%
Úplné počátky	Vaše nejvyšší hodnocení: 100%
Legie na Rusi	Zatím nevyplněný test

Obrázek 5 -Seznam dostupných testů

3.4. Průběh testu

Po zvolení libovolného dostupného testu uživatelem dojde ke spuštění souboru „LoadTest.jsp“. Tento soubor projde databázi a načte id všech dostupných otázek pro zvolený test, a ty přidá do pole typu ArrayList. To je posléze zamícháno a prvních pět otázek vypsáno uživateli jako samotný test. Tento způsob zaručuje obměnu otázek v testu a jejich pořadí, aby vyplňování testu nebylo založeno jen na memorizaci minulého testu. Vypsání se pak odvíjí podle typu otázky, pokud se jedná o otázku s jednou správnou odpovědí, vypíšu se odpovědi jako soubor vstupů typu „radio“. Pokud je to kategorie otázky více správných odpovědí, vypíše se skupina vstupů typu „checkbox“. Pokud se má jednat o časovou osu pak se vypíšu 4 okna s možností volby pořadí dané odpovědi a pokud se jedná o přiřazovací otázku, jedná se pak o tabulku se sloupci odpověď typu a, volba příslušné odpovědi typu b a odpověď typu b. Ke každé otázce se taky odesílá id a typ dané otázky pro pozdější kontrolu odpovědi. Samotná odpověď se ale během testu na stránce nenachází a nejde vyčíst ze zdrojového kódu. Každý element testu má nastavený atribut „name“ tak, aby nemohlo dojít k překrytí s dalšími otázkami, a aby fungovaly skupiny „radio“ a „checkbox“

Když je uživatel spokojen může kdykoliv odevzdat stiskem tlačítka „Odevzdat test“, tím se spustí soubor „MarkTest.jsp“, ten převezme data vyplněná na předchozí stránce a prvotně pomocí id otázky vyhledá v databázi řešení a podle druhu otázky jej společně s odpovědí načtených z testu upravuje, aby byly porovnatelné. Pokud je celá odpověď u otázky správná, je uživateli připsán bod a pokud ne, je vypsáno znění špatně zodpovězené otázky. Pokud uživatel na otázku neodpoví, je vynecháno porovnávání a program přechází k další odpovědi. Po kontrole všech odpovědí se vypíše koncové skóre, které se také uloží do databáze. Pokud již existuje v databázi záznam o

odpovědi a jeho skóre je nižší, proběhne úprava zápisu. Pokud neexistuje, je vytvořen nový záznam a tím je program hodnocení ukončen.

4. Lekce

Témata lekcí byla vybrána tak, aby pokryly vymezené časové období od počátku lidských dějin do konce druhé světové války. Jednotlivé lekce na sebe nijak nenavazují a nepředpokládají žádné předchozí znalosti. Samotná témata jsou zpracována především z knižních zdrojů, i když po vyhlášení nouzového stavu byly naše možnosti získání knižních zdrojů velmi omezeny. Tím jsou tedy lekce, které byly vytvořeny později, převážně z internetových zdrojů.

Inspirace pro témata lekcí většinou přicházela především z knih, které jsem někdy četl, nebo neznalost témat a zájem o prohloubení svých znalostí.

4.1. Vytváření lekcí

Možnost vytvoření nové lekce je v souboru „lekce.jsp“ v admin panelu pod záložkou Lekce. Zde je tlačítko „Přidat lekci“, které uživatele přesměruje do souboru „addNewLesson.html“¹⁴. K vytváření obsahu pro lekce, jsme použili textový editor TinyMCE. Tento editor může být jak placený, tak zadarmo. Skládá se totiž ze speciálních pluginů, z nichž jsou některé placené a jiné zadarmo, dá se z nich však poskládat seznam „free“ pluginů, díky kterým TinyMCE funguje jako dobrý textový editor, který umí skoro vše.

Premium pluginy jsou většinou pouze vylepšení free pluginů, takže potom fungují lépe, mají lepší funkce, je jednodušší s nimi pracovat. I z free pluginů se však dá poskládat slušný balíček, který poté funguje velmi dobře. TinyMCE editor se načítá místo tagu „textarea“, takže v addNewLesson.html je textové pole, do kterého se při načtení souboru načte TinyMCE editor se zvolenými pluginy přímo ze serverů <https://www.tiny.cloud/>, takže v našich souborech nezabírá žádné místo.

¹⁴ addNewLesson.html je jediným souborem .html v celém projektu, důvod je takový, že se TinyMCE nenačte v .jsp souboru.

Seznam využívaných pluginů

1. Code plugin - možnost editování přímo HTML kódu, který TinyMCE vytváří při psaní textu
2. Link plugin – možnost vkládání odkazů (<a>)
3. Lists plugin – možnost vytváření číslovaných a odrážkových seznamů
4. Media plugin – možnost přidávání videí a zvukových nahrávek
5. Paste plugin – tento plugin převádí do HTML kódu text zkopírovaný z Wordu a snaží se co nejlépe dodržet formát textu
6. Table plugin – možnost přidávat a upravovat tabulky
7. Autoresize plugin – postupně zvětšuje plochu editoru, jak uživatel píše obsah lekce, aby nemusel scrollovat
8. Autosave plugin – dá uživateli varování, pokud se pokouší opustit vytváření lekce, ale má v editoru napsaný nějaký obsah. Zároveň přidává možnost obnovit poslední verzi, pokud by uživatel například omylem zavřel vytváření lekce a chtěl by se vrátit
9. Character Map plugin – přidává tabulku se speciálními znaky pro lepší pohodlí, pokud člověk neví, jak znak napsat na klávesnici
10. Full Screen plugin – přidává možnost zvětšit TinyMCE editor na celou obrazovku
11. Image plugin – přidává možnost vkládat obrázky

Po kliknutí na tlačítko „Přidat lekci“ se otevře okno, kde je políčko se jménem „Název lekce“, sem uživatel zadá název lekce a z toho se také odvíjí název potom vytvořeného souboru .jsp. Když se vytváří lekce, vytvoří se soubor nazev_lekce.jsp, název lekce je ale upraven, všechna velká písmena jsou převedena na malá, diakritika odstraněna a mezery nahrazeny tímto znakem _.

Pokud bych chtěl vytvořit například tedy lekci s názvem Českoslovenké opevnění, název vytvořeného souboru bude ceskoslovenske_opevneni.jsp. Cesta, kde se vytváří soubor je předem daná: „/lekce/nazev_lekce.jsp“. Lekce jsou tedy vytvářeny ve složce lekce, aby se odlišily od dalších souborů.

Po názvu lekce se musí zvolit kategorie, do které bude lekce přidána, nyní máme čtyři možnosti volby: Pravěk, Středověk, Novověk a Československo, další kategorie se dají ale lehce přidávat, takže možnosti případného rozšíření jsou. Jako další a poslední políčko ve vytváření lekcí je už samotný TinyMCE editor, kam se píše obsah lekce, který TinyMCE převádí do HTML. Tento HTML kód se potom vloží do nově vytvořeného souboru a tím je nová lekce hotová.

Heroku nepovoluje úpravu souborů přímo v zapnuté instanci a vždy spustí „deploynutou“ verzi z Githubu. Z toho bohužel vyplývá, že na tomto hostingu se nedají vytvářet nové lekce, a proto jestliže chceme vytvořit novou lekci, tak si musíme „čejiny“ pustit na localhostu, vytvořit lekci tam, přidat záznam přidání do lokální databáze do Heroku databáze a „pushnout“ nově vytvořený soubor s lekcí na Github. Novou verzi „deploynout“ na Heroku a teprve potom budeme mít novou lekci i na Heroku doméně.

O samotné vytvoření lekce se stará soubor „lekceCreated.jsp“, který získá všechny zadané údaje o lekci ze souboru „addNewLesson.html“ a přetvoří je do souboru .jsp. Nastaví si název souboru, cestu a obsah lekce. Potom se pokusí vytvořit soubor, a když se mu to podaří, pokračuje na zápis do databáze¹⁵, kam zapíše novou lekci, a nakonec vypíše uživateli, že se podařilo vytvořit lekci. Je ale také možné, že uživatel zadá stejný název lekce, jaký už existuje. V tomto případě vypíše uživateli, že se nepodařilo vytvořit novou lekci a uživatel má možnost vrátit se do admin panelu.

4.2. Vypsání lekcí

Uživatel se na domovské stránce může dostat k lekcím v záložce „Lekce“, kde si může zvolit buď seznam kategorií, anebo seznam všech lekcí. Jakmile si zvolí výpis podle kategorií, je přesměrován na stránku „KategorieLekce.jsp“, kde je seznam všech kategorií. Zde si uživatel zvolí například kategorii „Československo“ a je přesměrován na stránku „ceskoslovensko.jsp“, která se připojí k databázi, vyhledá všechny zápisy s touto kategorií a vypíše je uživateli do listu na obrazovce.

¹⁵ Zapisuje name, path a category, ostatní se vytvoří samo, id je nastaveno jako serial, takže se vytváří samo a test se automaticky nastaví na hodnotu false.

Aby se dalo kliknout kamkoliv do listu a nemuselo se vždy klikat jen na nápis názvu lekce, tak je do tohoto výpisu přidán JavaScriptový soubor „clickableList.js“, který obstará, aby si soubor „ceskoslovensko.jsp“ myslel, že jakmile uživatel klikne vedle do listu, že kliknul přímo na odkaz.

Výpis lekcí jsou jednoduše odkazy na soubory .jsp vytvořené v admin panelu. TinyMCE vytváří HTML kód, takže když je vytvořen soubor nazev_lekce.jsp, tak už stačí jen se na něj v listu odkázat.

Podobně to funguje ve výpisu všech lekcí, jen soubor „SeznamLekci.jsp“ vyhledá v databázi všechny záznamy o lekcích a vytvoří na ně odkazy v listu. Uživatel si tudíž už jen klikne na lekci, jakou chce a může si přečíst něco o dějinách našeho národa.

5. Závěr

Náš projekt jsme tvořili podle zadání, ale při vývoji jsme narazili na několik komplikací, které jsme museli řešit jinak, než jsme očekávali při návrhu zadání. Největším problémem, který nás zdržel, byla nevhodná volba hostingu, ale tu jsme později vyřešili přechodem na Heroku. V zadání jsme také uváděli, že budou otázky i lekce uloženy v .txt souborech. Postupem času jsme celou sekci otázek přesunuli do databáze pro lepší výběr samotných otázek, který nám umožnil náhodné pořadí otázek v testu. Lekce jsme také přemístili, ale jen částečně. Lekce mají záznam jak v databázi, tak i soubor .jsp se samotným textem, který není vhodný pro ukládání v databázi. Jinak jsme se pokusili co nejlépe splnit zadání a přidat i věci navíc.

Jako možnosti vylepšení vidíme samozřejmě vytvoření lekcí, především pak o ty, které jsou omílané všude jinde jako jsou doby Karla IV., husité a podobně, kterým jsme se vyhnuli právě z důvodu jejich obecné známosti, s čímž by souviselo i vytváření dalších testů. V současné verzi jde vytvářet nové lekce přes administrátorský účet, na který běžní uživatelé nebudou mít přístup. Nové lekce a testy tedy mohou být vytvářeny pouze námi, což značně omezuje možnosti rozšiřování tohoto projektu.

6. Bibliografie

- Minh, N. (28. Červen 2019). *Codejava.net*. Načteno z How to implement authentication filter for Java web application: <https://www.codejava.net/java-ee/servlet/how-to-implement-authentication-filter-for-java-web-application>
- Minh, N. (4. Červenec 2019). *Codejava.net*. Načteno z How to code login and logout with Java Servlet, JSP and MySQL: <https://www.codejava.net/coding/how-to-code-login-and-logout-with-java-servlet-jsp-and-mysql>
- Refsnes Data. (2020, Leden 1). *W3.CSS Home*. Retrieved from w3schools.com: <https://www.w3schools.com/w3css/default.asp>
- Wikimedia Foundation. (19. Květen 2019). *Apache Tomcat*. (Wikimedia Foundation) Získáno 29. Duben 2018, z Wikipedia: https://cs.wikipedia.org/w/index.php?title=Apache_Tomcat&oldid=17255822
- Wikimedia Foundation. (22. Leden 2020). *Heroku*. (Wikimedia Foundation) Získáno 30. Duben 2018, z Wikipedia: <https://en.wikipedia.org/w/index.php?title=Heroku&oldid=937021768>
- Wikimedia Foundation. (13. Duben 2020). *JavaServer Pages*. (Wikimedia Foundation) Získáno 29. Duben 2018, z Wikipedia: https://cs.wikipedia.org/w/index.php?title=JavaServer_Pages&oldid=18389958

7. Seznam obrázků

Obrázek 1 - Google tlačítko k přihlášení	3
Obrázek 2 - tabulka lekcí v admin panelu	5
Obrázek 3 - Volba editace otázky pod daným tématem.....	7
Obrázek 4 - Volba samotné otázky pro editaci	7
Obrázek 5 -Seznam dostupných testů	8

8. Přílohy

8.1. Seznam literatury použité na tvorbu lekcí

2008. *Magnae Moraviae fontes historici*. Brno : Ústav klasických studií, Masarykova univerzita, Filozofická fakulta, 2008. 978-80-254-1780-5.

Biegel, Richard. *Přehled dějin českého umění - Renesanční architektura v českých zemích*. [Prezentace] Praha : Filozofická fakulta Univerzity Karlovy v Praze.

Bláhová, Marie. 2005. *Kronika tak řečeného Dalimila*. Praha : Paseka, 2005. 80-7185-767-X.

Daubner, Petr. 2019. Třicetiletá Válka 1618-1648 – Falcká, Švédská a Dánská válka, Obnovené zřízení zemské. *Skompase.cz*. [Online] 11. listopad 2019. [Citace: 17. duben 2020.] https://skompasem.cz/tricetileta-valka-1618-1648/#Cesko_8211_Falcka_Valka_1618-1623.

Keltské rotační mlýny v Čechách. **Waldhauser, Jiří. 1981.** 72, Praha : nakladatelství Academia, 1981.

Ladislav, Čermák. 2008. Bunkry.cz - Historie československého opevnění z let 1935-38. *Bunkry.cz*. [Online] 25. listopad 2008. [Citace: 20. duben 2020.] <https://www.bunkry.cz/clanek/1180>.

2010. Laténská doba - mladší doba železná - *archeolog.cz*. *Archeolog.cz*. [Online] 16. 1 2010. [Citace: 15. 4 2020.] <http://www.archeolog.cz/encyklopedie/latenska-doba/>.

Lutovský, Michal. 2006. *Bratrovrah a tvůrce státu*. Praha : Set out, 2006. 80-86277-49-6.

Obrazová, Pavla a Vlk, Jan. 1994. *Maior Gloria - svatý kníže Václav*. Praha : Paseka, 1994. 80-85192-94-2.

Pitro, Martin a Vokáč, Petr. 2018. *Průvodce třicetiletou válkou*. Praha : Universum, 2018. 978-80-242-6063-1.

Plus, Historie. 2019. *Nezávislé Československo se zrodilo v Rusku. Legionáři vydobyli republiku skvělou pověst*. [Podcast] Praha : Český Rozhlas plus, 2019.

Smetana, Mirek. 2008. *Renesance v českých zemích*. [Prezentace] Brno : Masarykova univerzita v Brně, 2008.

Tesař, Jan. 2000. *Mnichovský komplex*. Praha : Prostor, 2000. 80-7260-035-.

Vácha, PhDr. Dalibor. 2011. *Život v legiích*. České Budějovice : Jihočeská univerzita v Českých Budějovicích Historický ústav Filozofické fakulty, 2011.

Zlatohlávek, Martin. *Přehled dějin českého umění - Sochařství a malířství renesance v Čechách*. [Prezentace] Praha : Filozofická fakulta Univerzity Karlovy v Praze.