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

NEVIM

Jméno a příjmení: Karpiš, Jonáš
Třída: 4.A
Studijní obor: Informační technologie 18-20-M/01
Školní rok: 2023/2024

Zadání maturitní práce

Jméno a příjmení:	Jonáš Karpiš
Školní rok:	2023/2024
Třída:	4. A
Obor:	Informační technologie 18-20-M/01
Téma práce:	*Webová aplikace na procvičování VIM*
Vedoucí práce:	*RNDr. Jan Koupil, Ph.D.*

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

Cílem projektu je vytvořit webové prostředí pro výuku ovládání programu VIM hravým způsobem. Mimo přeskočitelného základního tutoriálu budou dostupná tréninková cvičení.

Obsahem budou tutoriálová cvičení, která prochází jednotlivé prvky VIMu, a také obecná cvičení, která spojí učivo z tutoriálů. Architektura bude umožňovat snadné přidávání dalších tutoriálů a cvičení.

Tutoriály budou sestaveny podle obtížnosti a každé tutoriálové cvičení bude mít dostupnou nápovědu. Vypracovaná cvičení bude aplikace vyhodnocovat, ukládat status a čas zhotovení.

Cvičení budou plně nezávislá, tak aby si uživatel mohl volit cvičení libovolně nebo je opakovat. U každého cvičení bude tabulka nejlepších dosažených výsledků.

Základ pro realizaci projektu bude tvořit JavaScriptový port editoru VIM nad kterým bude celé prostředí vybudováno. Součástí prostředí bude i systém uživatelských oprávnění členěných takto:

- Nepřihlášený uživatel – Má přístup ke všem tutoriálům a cvičením, jeho informace se neukládají, pouze dočasně
- Přihlášený uživatel – Má přístup ke všem tutoriálům a cvičením, jeho informace se ukládají a vyhodnocují, také může předložit návrh na nové cvičení.
- Moderátor – Má možnost přidávat nová cvičení a potvrzovat návrhy cvičení od uživatelů.
- Admin – Spravuje moderátory

Časový harmonogram:

- **Září:** Analýza problému, návrh konceptů cvičení, tvorba UML usecase diagramů, návrh vzhledu stránky, výběr vhodných technologií
- **Říjen – listopad:** Tvorba uživatelského rozhraní, návrh realizace databáze a backendu
- **Prosinec:** Tvorba a finalizace backendu, deploy na server
- **Leden – únor:** Testování, dokončovací úpravy
- **Březen:** Práce na dokumentaci

Poděkování

Chtěl bych poděkovat panu učiteli RNDr. Janu Koupilovi Ph.D. za umožnění práce na maturitním projektu pod jeho vedením. Také bych chtěl poděkovat mým kamarádům, kteří mi pomohli s tvorbou aplikace.

Prohlášení

Prohlašuji, že jsem projekt vypracoval samostatně a využil jsem pouze prameny a reference uvedené v seznamu bibliografických záznamů.

Klíčová slova

Vim, Editor, Monaco, cvičení, level

Abstrakt

Práce dokumentuje návrh a vývoj webové stránky Nevim, na které se nachází cvičení pro trénink ovládání textového editoru Vim. Realizace projektu zahrnuje backendovou část, návrh cvičení a webový frontend s ovládáním cvičení. V řešení je využito kódového editoru Monaco Editor od Microsoftu.

Obsah

Úvod 6
1. Existující řešení 7
1.1. Vimtutor 7
2. Výběr technologií 8
3. Realizace 9
4. Způsoby řešení 12
6. Bezpečnost 16
Závěr 17
Reference 18

Úvod

Programování je neodmyslitelně spojeno s psaním kódu. Teoreticky je možné psát kód i v nejjednodušším z textových editorů, existuje celá řada speciálních nástrojů zaměřená na programátory.

Jedním z dnes používaných textových editorů je také Vim, neboli Vi Improved, který je často nazýván „editorem programátorů“ [1]. Ten je populární mezi vývojáři díky své efektivitě při psaní kódu. Bohužel ale odrazuje nováčky, neboť je poměrně strmá jeho učící křivka.

Vim je modální textový editor, funkční ve dvou ‘módech’ – Insert a Normal s primárně textovým rozhraním, neboli funkční v příkazovém řádku. Tento způsob upravování textových souborů se zde neobjevil poprvé, ale byl převzat od editoru Vi, ze kterého vychází. Vim také nabízí další funkce, jako například persistentní multi-level ‚undo‘ strom, extenzivní systém na pluginy a podporuje mnoho programovacích jazyků. [1]

Tento projekt dokumentuje vývoj webové aplikace určené pro trénink využívání textového editoru Vim.

1. Existující řešení

1.1. Vimtutor

Vimtutor je interaktivní tutoriál vytvořen přímo vývojáři Vimu. Příkaz „vimtutor“ se běžně objevuje v unixových systémech. Po spuštění příkazu se v terminálu otevře textový soubor se zadáním pro daný úkol. Vimtutor má za cíl nechat uživatele pracovat samostatně na jednotlivých úkolech. Tím je dokázáno, že se uživatelé neučí daná cvičení zpaměti. Celý tutoriál zabere uživateli 25-30 minut a poté bude umět ovládat Vim se základními funkcemi. [2]

1.1. Openvim.com

Webová stránka Openvim.com je také interaktivním tutoriálem. Nabízí uživateli velmi přímočarý přístup ke způsobu učení Vimu. Tutoriál velmi podrobně poskytuje instrukce při plnění jednotlivých cvičení. Aplikace simuluje vzhled počítače, ve kterém uživatel dostane úkol pro postup v tutoriálu.

1.2. Vim-adventures.com

Dalším tutoriálem je webová stránka Vim-adventures.com, která se na vzdělávání se v programu Vim dívá z jiného úhlu pohledu. Nabízí rozhraní 2D světa, ve kterém hráč interaguje s okolním světem a pohybuje se pomocí Vim příkazů a zkratek. [3]

2. Výběr technologií

2.1. React

React je JavaScriptová knihovna užitečná pro stavbu komponentů webových stránek a usnadňuje tvorbu uživatelského rozhraní. React byl vyvinut společností Meta, byl vydán v roce 2013 a aktuálně funguje na verzi 18. [4]

2.2. Next.js

Next.js (dále Next) je React framework užívaný pro vývoj full-stack webových aplikací. Next má mnoho funkcí, například "server-side rendering", statické stránkování a mnoho jiných funkcionalit pro optimalizaci. React vytváří rozhraní s komponenty a Next dodává dodatečné funkce. [5]

2.3. Firebase

Pro databázi se využívá Firebase od firmy Google. Nabízí velké množství služeb, například autentikaci a integraci různých aplikací, ale pro naše užití je potřebná pouze databáze.

2.4. NextAuth.js

Knihovna NextAuth.js zajišťuje přihlašování a autentikaci aplikace. Tato knihovna pro Next.js umožňuje snadnou autentikaci bez ukládání dat do databáze. Také usnadňuje nasazení jakékoliv OAuth služby, v našem případě Google.

2.5. Monaco Editor

Pro samotnou strukturu cvičení se využívá open source kódový editor Monaco od Microsoftu. Monaco Editor je primárně známý z editoru VS Code. Také nabízí veřejnou knihovnu pro Next.js.

3. Realizace

3.1. Cíl projektu

Projekt Nevim byl vytvořen s myšlenkou snadného a interaktivního výukového programu pro usnadnění používání Vimu pro nové uživatele. Je to webová aplikace s tutoriálem, která provede nováčka základními funkcemi Vimu pomocí interaktivních cvičení.

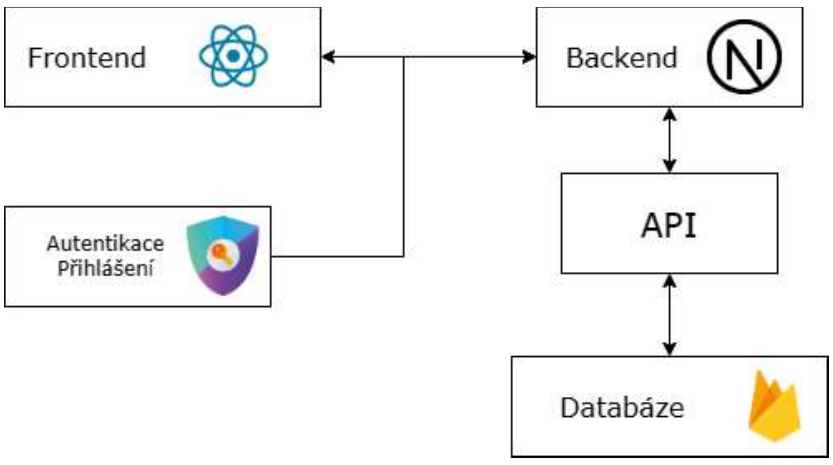
Stavba a průběh cvičení v aplikaci jsou inspirovány postupem cvičení ve skriptu vimtutor. Ten prochází základy Vimu, již dříve zmíněn v kapitole o existujících řešeních.

Projekt obsahuje uživatelské prostředí, jehož částí je webová stránka, na které jsou samotná cvičení ve Vimu. Cvičení jsou vytvořena pro uživatele takovým způsobem, aby ho odnaučila špatným návykům při používání Vimu a naučila ho být efektivnější v práci s editorem.

Aplikace si ukládá pokrok uživatelů v průběhu cvičení, aby při příštím připojení mohli pokračovat ve svém procvičování. Také své výsledky mohou porovnávat s dřívějšími. Aplikace si taktéž ukládá čas zhotovení jednotlivých cvičení, aby v případě, že uživatel všechny cvičení zhotoví, mohl stále vylepšovat své dovednosti v užívání Vimu.

3.2. Architektura

Cílem práce je vytvořit webovou aplikaci, na které se nachází rozhraní pro uživatele, ve kterém může plnit cvičení v prostředí Vim. Admin aplikace má přístup do rozhraní, kde má možnost přidávat další cvičení. Dále je potřeba ukládání dat samotných cvičení, uživatelů a jejich postupu v tutoriálu.



Obrázek 1: Architektura aplikace

Z těchto pokynů vychází základní architektura projektu s rozdělením na prvky:

3.2.1. Frontend

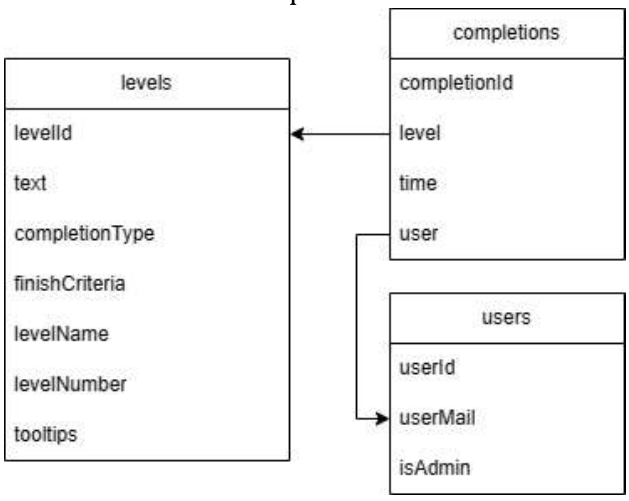
Uživatelské rozhraní je vytvořeno s pomocí Reactu. Využívají se takzvané komponenty pro snadné skládání stránky. Frontend připravuje stránku pro uživatele a zařizuje komunikaci mezi ním a aplikací.
Na cvičení je využíván Monaco Editor, na kterém je nasazen plugin Monaco Vim. Tato kombinace vytváří prostředí, ve kterém uživatel může plnit cvičení a aplikace kontroluje jeho vstupy při řešení.

3.2.2. Backend

Celý projekt závisí na backendu tvořeným s frameworkem Next.js. Tento prvek se stará o správný běh aplikace, zpracování cvičení, ukládání a čtení dat z databáze.
Backend přijímá uživatelské vstupy z frontendu a zároveň dodává důležité informace a data, například cvičení či způsob řešení.
Tato data jsou uložena v databázi, se kterou backend komunikuje pomocí API (Application programming interface).
Backend připraví request, který API vyžádá od databáze. Po přijetí požadavku databáze vrátí odpověď, kterou přijme API a dále přepošle na backend pro následující užití.

3.2.3. Databáze

Databáze je pro projekt klíčovým prvkem. Jsou v ní uložena veškerá data, se kterými backend pracuje. Aplikace využívá služby Firebase od Googlu, jelikož dodává rychlé, bezpečné a spolehlivé hostování databáze.
Do databáze se ukládají jak cvičení, tak přihlášení uživatelé a uživatelské zhotovení cvičení. Data jsou následovně připravena na užití v aplikaci při dalších operacích.



Obrázek 2: Schéma databáze

3.2.4. Autorizace a přihlašování

Přihlašování a autentizaci zajišťuje knihovna NextAuth.js, jelikož funguje na systému session tokenů pro zvýšenou bezpečnost. Knihovna dovoluje snadné nasazení pro veřejné autentizační systémy, například Google, což je hlavní způsob přihlášení do aplikace.

4. Způsoby řešení

4.1. Cvičení

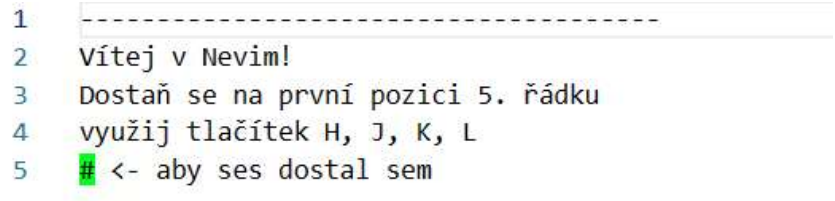
Nejdůležitější částí aplikace jsou cvičení, která mají za úkol uživateli vysvětlit Vim a naučit ho pracovat v editoru efektivně. Ze cvičení je vytvořen tutoriál, který zaručí uživateli logický průvod všemi základními funkcemi Vimu. Pro budoucí užití je možné adminem tutoriál upravovat přidáváním či mazáním cvičení.

4.1.1. Struktura

V aplikaci jsou dva druhy cvičení:

- **Pozice kurzoru** - cvičení je vyhodnoceno za správné, pokud se uživatel dostane na správnou pozici kurzoru v textu.
- **Editace textu** - cvičení je zhotoveno, když uživatel upraví text tak, aby se shodoval se zadáním.

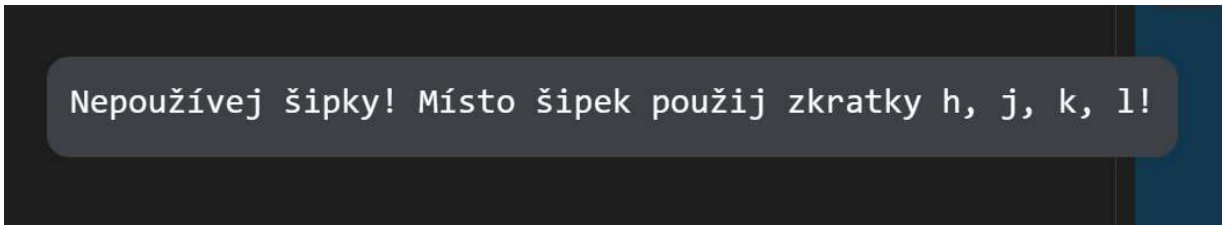
Úkol je zadán v textu cvičení, ve kterém ho uživatel vyplňuje. Tento způsob napodobuje přístup učení ve Vimtutoru, jelikož podporuje volný průchod tutoriálem a učí uživatele používat Vim vlastním tempem.



Obrázek 3: Ukázka cvičení v aplikaci Nevim

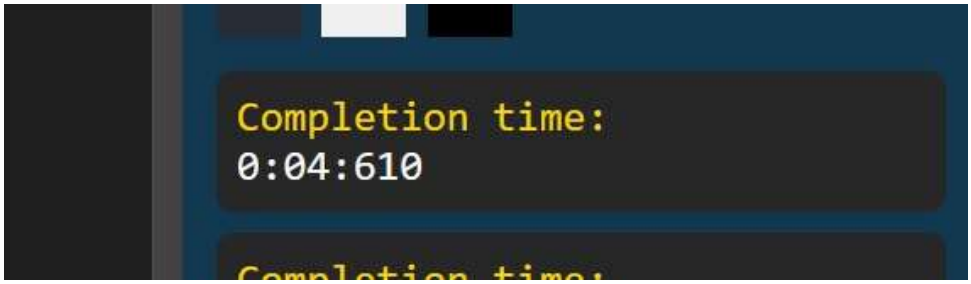
4.1.2. Vyhodnocování a upozorňování

Při vyplňování cvičení si aplikace kontroluje uživatelovu práci, a upozorňuje jej na neefektivní způsoby řešení. Při tvorbě nového cvičení si tvůrce vybírá, které zkratky budou spouštět upozornění o plnění úkolu. Upozornění jsou mírná a nepřikazují uživateli přímo jak cvičení plnit. Spíše jsou tipy odrazující špatné návyky, viz. Obrázek č. 4.



Obrázek 4: Upozornění uživatele při cvičení

Když aplikace uzná cvičení za zhotovené, vypíše čas zhotovení a následovně uloží data do databáze s informací, kdo zhotovil jaké cvičení a za jaký čas. Tato funkce dává uživateli možnost vidět svůj pokrok v tutoriálu a také se porovnávat s předešlými výsledky.



Obrázek 5: Zhotovené cvičení uživatele

Samotná cvičení jsou veřejná, tudíž jsou přístupná i pro nepřihlášené uživatele. Aplikace si ale ukládá pokrok v učení pouze přihlášených účtů.

4.1.3. Postup

Pořadí cvičení je inspirováno posloupností ve Vimtutoru, která prochází všechny základní prvky Vimu tak, aby na sebe smysluplně navazovaly.

Zjednodušený postup tutoriálu prochází příkladně:

- Základní pohyb v souboru – zkratky h, j, k, l
- Editace textu – připisování do textu, mazání jednotlivých znaků, slov a řádků v textu
- Používání operátorů – pohyb o daný počet slov, mazání řádku, aj.
- Příkaz undo
- Vyhledávání a nahrazování
- Změna písmene, slova, řádku
- Zkratka Ctrl+G

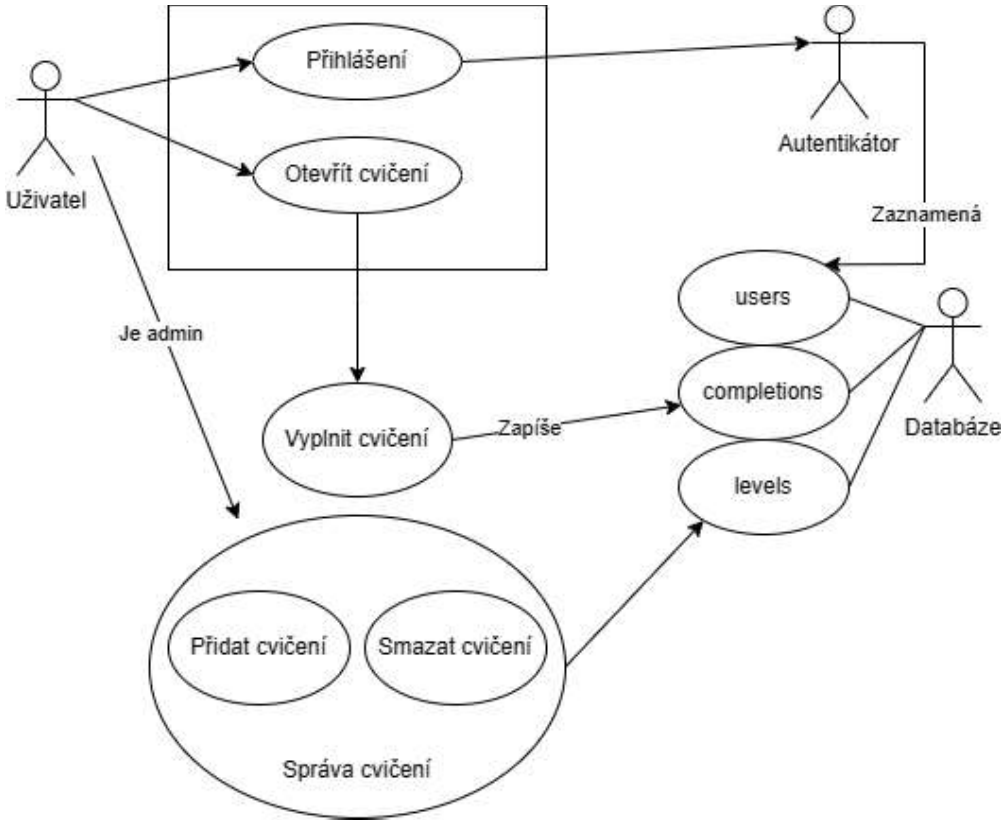
Cvičení jsou sepsána v tomto pořadí pro docílení snadného průběhu tutoriálem. Uživatel se nejprve učí pohyb v souboru, následně se dozvídá o editaci textu a dostává se tak k mírně složitějším kombinacím zkratek v editoru.

4.1.4. Správa cvičení

Uživatel přihlášen pod účtem admin má z domovské stránky přístup na rozhraní pro správu cvičení v tutoriálu. V tomto rozhraní má uživatel možnost odebrat a přidat nová cvičení. Pro přidání nového cvičení je přítomen formulář, ve kterém tvůrce vyplňuje název, druh a text cvičení a také pomocné upomínky na špatné způsoby řešení.
Po potvrzení přidání cvičení aplikace zašle požadavek na backend, aby uložil cvičení do databáze.

4.2. UML Diagram

Use case diagram zobrazující případy užití:



Obrázek 6: UML Diagram aplikace

5. Bezpečnost

Zabezpečení je důležitá část jakéhokoliv projektu a kvůli ochraně uživatelských dat je potřeba zajistit bezpečnost.

Hlavní zranitelností aplikace je neautorizovaný přístup do správy cvičení, kde se nacházejí různé nástroje pro úpravu dat. Autentizaci a přihlašování v projektu zajišťuje NextAuth.js, což je React knihovna, která umožňuje kontrolu, zda je uživatel přihlášen při každé akci.

Závěr

V rámci projektu byla navržena a sestavena webová aplikace Nevim. Aplikace je vytvořena s prostředím pro výuku ovládání textového editoru Vim interaktivním způsobem. Projekt obsahuje backendovou část a webový frontend, na kterém se zobrazují cvičení.
Projekt je nasazen pomocí služby Vercel na adrese nevim-green.vercel.app.

Prací na projektu jsem se naučil lépe ovládat editor Vim a také práci s kódovým editorem Monaco. Naučil jsem se navrhnout a vytvořit webové rozhraní s úkoly v editoru Vim a rozhraní pro správu cvičení.

Reference

[1] Vim, „vim.org,“ 2007. [Online]. Available: https://www.vim.org/about.php. [Přístup získán 14 3 2024].

[2] R. G. R. Suero, „vimschool.netlify.app,“ Hugo, [Online]. Available: https://vimschool.netlify.app/introduction/vimtutor/. [Přístup získán 14. 3. 2024].

[3] D. Linder, „https://vim-adventures.com/,“ 2012. [Online]. Available: https://vim-adventures.com/. [Přístup získán 21. 3. 2024].

[4] Meta, „https://react.dev/,“ Meta Open Source, 2024. [Online]. Available: https://react.dev/. [Přístup získán 19. 3. 2024].

[5] Vercel, „https://nextjs.org/,“ Vercel, Inc., 2024. [Online]. Available: https://nextjs.org/docs. [Přístup získán 19. 3. 2024].