



Gymnázium Arabská, Praha 6, Arabská 14

Obor programování, vyučující Mgr. Jan Lána



Jako Pavouk

Psaní všemi deseti

Filip Růžička

Duben, 2024

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 dne

Filip Růžička

Anotace

Zadání projektu

Webová aplikace na výuku psaní všemi deseti. Bude se skládat z lekcí přidávajících vždy 2-3 písmena na klávesnici. Tímto způsobem si člověk postupně osvojí celou klávesnici. V každé lekci budou cvičení s náhodnými písmenky, ale i se slovy obsahující pouze již naučená písmena.

Aplikace bude taktéž obsahovat přihlašovací systém, díky kterému bude mít každý uživatel k dispozici profil pro zobrazení svých zlepšení a statistik jako např. průměrnou rychlost, chybovost, procento dokončených lekcí atd. Co se týče technologií, použiji programovací jazyk Go pro backend, framework Vue.js pro frontend a databázi PostgreSQL pro ukládání dat o uživateli a pro přístup k slovníku spisovné češtiny.

Abstract

Project assignment

A web application for learning to write with all ten fingers. It will consist of lessons adding 2-3 letters each time. In this way, one will gradually master the entire keyboard. In each lesson there will be exercises with random letters, but also with words containing only already learned letters.

The app will also include a login system, which will allow each user to have a profile to view their improvements and statistics such as average speed, error rate, percentage of lessons completed, etc. In terms of technology, I will use the Go programming language for the backend, the Vue.js framework for the frontend, and the PostgreSQL database for storing user data and for accessing the Czech spelling dictionary.

Obsah

1. Úvod	1
1.1. Výběr tématu	1
1.2. Struktura	1
1.3. Použité technologie	1
1.3.1. Frontend	1
1.3.2. Backend	2
1.3.3. Databáze	2
1.3.4. Dokumentace	2
2. Psaní všemi deseti	3
2.1. Rozložení klávesnice	3
2.2. Prstoklad	4
2.3. Implementace	4
3. Problematika	5
3.1. Slovník	5
3.2. Fungování na mobilních zařízeních	5
3.3. Dvě různá rozložení klávesnice	5
3.4. Optimalizace	7
3.5. Výběr jmen uživatelů	7
4. Design	8
4.1. Ilustrace	8
4.2. Barevné schéma	8
5. Další vývoj aplikace	9
5.1. Statistika mezi uživateli	9
5.2. Systém pro školy	9
5.3. Věty pro lekce	9
6. Závěr	10
Odkazy	11

1. Úvod

1.1. Výběr tématu

Nápad na vývoj této webové aplikace souvisel s faktem, že já sám **jsem neuměl psát všemi deseti**, a tak jsem se to rozhodl naučit. Se zděšením jsem ale zjistil, že co se týče české klávesnice, není moc možností, přičemž nějaké z nich stojí skoro 1000 Kč.

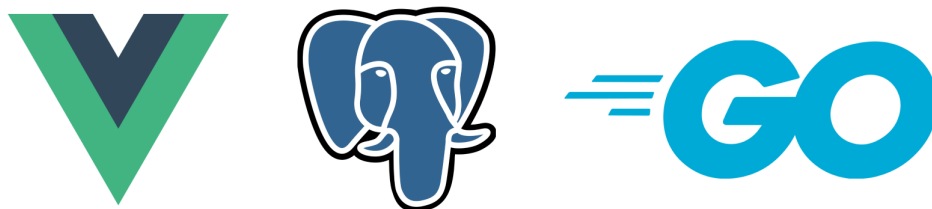
1.2. Struktura

Můj projekt se skládá ze 3 hlavních částí. První částí je takzvaný **frontend**. To je ta část projektu, kterou vidí a s kterou interaguje uživatel. V mém případě do ní patří například design, statistika nebo samotné psaní textů.

Druhou částí je takzvaný **backend** neboli program běžící na serveru. Tuto část aplikace sice nemá uživatel možnost vidět, ale neustále s ním komunikuje. Můj backend se stará o autentifikaci uživatelů (registrace, přihlašování, tokenizace, změna hesla), o posílání ověřovacích e-mailů a generování textů na psaní.

Poslední částí je **databáze**, která ukládá data o uživateli, disponuje seznamem lekcí, slovníkem a dalšími daty potřebnými k chodu aplikace.

1.3. Použité technologie



Obrázek 1: Vue.js¹, PostgreSQL², Go³

1.3.1. Frontend

Pro vývoj frontendu neboli samotné webové stránky jsem si vybral framework **Vue.js**. Hlavním důvodem bylo, že už jsem s ním měl zkušenosti ze skupinového projektu pro minulý rok.

Místo Javascriptu jsem používal **Typescript**, ale nejsem si jist, jestli ho mohu doporučit. Čas od času jsem měl pocit, že musím psát hodně kódu navíc a že to přináší **více škody než užitku**.

Co se týče kaskádových stylů, držel jsem se **čistého CSS**, protože si myslím, že s nástroji jako Bootstrap nebo Tailwind ztrácíte jak čitelnost, tak kontrolu nad kódem. Přijde mi, že CSS se odděluje od samotného HTML z nějakého důvodu a nechci se vracet k tomuto:

```
<ul style="display: flex; justify-content: center;">
  <li style="width: 125px; height: 50px; transition:
    background-position-x 0.9s linear; text-align: center;">
    <a href="/" style="font-size: 22px; color: #777;
      text-decoration: none; transition: all 0.45s;">Home</a>
  </li>
  <li style="width: 125px; height: 50px; transition:
    background-position-x 0.9s linear; text-align: center;">
    <a href="/about" style="font-size: 22px; color: #777;
      text-decoration: none; transition: all 0.45s;">About</a>
  </li>
</ul>
```

1.3.2. Backend

Jako jazyk pro backend, program starající se o data, jsem si zvolil **Go**, protože se mi v té době hodně líbil Python a chtěl jsem zkusit něco podobného, ale více low-level. Dnes se mi ale zpátky nechce.

Použil jsem také framework Fiber, i když teď cítím, že bych se v klidu obešel bez něj, pouze za pomoci **standardní knihovny**.

Ještě jsem totiž nevěděl, jak dobrou standardní knihovnu Go má a že mindset Go programátorů je úplným opakem JS developerů. Když chce člověk něco udělat v JavaScriptu, většinou už existuje asi 287 knihoven, které všechny dělají to samé trochu jinak. Nějaká lépe, nějaká hůř. Vybereme si tedy jednu z nich a začneme bezhlavě importovat cizí programy, kterým nerozumíme a ani nechceme rozumět.

V Go to tak není. Cokoliv, co potřebujete od regulárních výrazů přes unit testování, datum a čas až po komunikaci se sítí už je v standardní knihovně, psané lidmi, kteří rozumí nejen jazyku samotnému, ale i danému problému.

1.3.3. Databáze

Pro uchovávání dat o uživateli, slovech pro psaní a podobně jsem zvolil relační databázi **PostgreSQL**, kvůli její popularitě a výkonu.

Ve výběrovém konkurzu byly i alternativy jako MongoDB nebo SQLite. Mongo jsem ale zavrhl, protože jsem chtěl spíše relační SQL databázi a SQLite je, podle mého názoru, lepší pro projekty v menším měřítku.

1.3.4. Dokumentace

Pro psaní této dokumentace jsem využil **Typst** „A markup-based typesetting system that is designed to be as powerful as \LaTeX while being much easier to learn and use.“⁴ Chtěl jsem se vyhnout stylování v MS Word, a \LaTeX mi přišel dost zkostnatělý. Chvilku jsem tedy hledal a nakonec zkusil moderní, také open-source, alternativu.

2. Psaní všemi deseti

2.1. Rozložení klávesnice

Klávesnice, kterou používáme každý den na počítači nebo telefonu, není vůbec uzpůsobena na rychlé ani pohodlné psaní. Proč to tak ale je? Proč používáme zrovna rozložení QWERTZ a QWERTY? Proč ne třeba ABCDEF?

Všechny cesty vedou až do roku 1878 k **psacímu stroji** Remington⁵, který se objevil s nám známou QWERTY a o deset let později bylo toto rozložení dokonce přijato za standard. Konstrukteři se při vytváření klávesnice k prvním psacím strojům totiž potýkali s **problémem zasekávajících se kladívek** stisknutých rychle za sebou. Docházelo k tomu ještě častěji, pokud byly tyto dvě klávesy vedle sebe.⁶ Proto bylo nutné klávesy, které se často píší za sebou sobě, „rozházet“ dál od sebe a vznikla tak nám známá QWERTY.

V Česku nebo například v Německu se převážně používá lehce modifikovaná QWERTZ, jednoduše proto, že v **němčině** se písmeno Z vyskytuje daleko častěji než Y a tak si tyto klávesy Němci prohodili, aby měli Z blíže po ruce. V té době se na našem území psalo převážně německy, a tak jsme tuto úpravu přijali.⁶

Přestože dnes už takový problém s kladívkem nemáme, zvyk je železná košile.



Obrázek 2: Psací stroj⁷

2.2. Prstoklad

Když už víme, jak vznikla naše klávesnice, pojďme se podívat, jak rozmístit prsty, abychom ji celou ovládli. Záchranným bodem pro nás jsou klávesy F a J ležící téměř ve středu. Ty na drtivé většině klávesnic mají **malé výstupky**, hmatné i poslepu. Na tyto dvě hlavní umístíme ukazováčky a ostatní prsty na klávesy ve stejné řadě. Vycházet tedy budeme z tlačítek ASDF a JKLŮ. Každý prst si poté hlídá svůj **pomyslný sloupeček**, jak můžete vidět na obrázku 3. Výjimkou jsou palce. Ty se při psaní starají pouze o mezerník.



Obrázek 3: Prstoklad

2.3. Implementace

Některé starší programy na výuku psaní všemi deseti fungovaly na bázi opisu textu o řádek níže. Poté porovnávají zadání s přepisem od uživatele. Tento způsob ale vede k častému dezorientování a k námaze očí z ustavičného koukání tam a zpět.

```
ffjj jjff fjfj fjff fffj jfff    <- zadání  
ffjj jjff fj|                    <- přepis
```

Jako Pavouk k psaní přistupuje trošku jinak. Uživatel text neopisuje, ale vše se děje přímo před jeho očima. Přesněji se uživatel orientuje podle podtržení, které označuje vždy další písmeno, které má stisknout. Text již napsaný je upozaděn šedým odstínem písma, aby se uživatel soustředil pouze na slova nadcházející, která jsou bílá. Špatné písmeno je značeno červenou barvou a je na něj upozorněno zvukem.

Moje aplikace mimo jiné monitoruje nejen rychlost psaní, přesnost nebo čas, ale i písmena, ve kterých uživatel nejčastěji chybí. Tato statistika je mu pak samozřejmě k nahlédnutí.

3. Problematika

3.1. Slovník

Jednou z prvních komplikací bylo najít nějaký dobrý slovník (respektive seznam českých slov, ne jejich významy). Jediný takový seznam, který jsem byl schopen dohledat, byl Český národní korpus SYN2015 o cca 70 000 slovech, který jsem si zatím zredukoval asi na **61 000** a stále čas od času narazím na nějaké cizí nebo zvláštní slovo. Ručně projít tisíce slov totiž není žádný med.

Věty objevující se v testu psaní jsem posbíral z webu Pohádkozem⁸. Jejich pohádky jsou ze starších knih, jejichž autoři už jsou více než 70 let po smrti.

3.2. Fungování na mobilních zařízeních

Fungování aplikace na výuku psaní všemi deseti na mobilních zařízeních je docela **paradox**. Přestože samotné psaní nedává smysl na malém displeji, musí být stránka přístupná na jakémkoli stroji. V roce 2023 asi 3/4 návštěvnosti webu totiž pocházelo z telefonů.⁹

Jako Pavouk je tedy responzivní a plně funkční na malých displejích, co se obsahu, přihlašovacího systému, statistik a dalšího týče. Když se ale uživatel pokusí vstoupit do samotného přepisování textů, setká se s hláškou „*Psaní na telefonech zatím neučíme...*“

Aplikace detekuje mobilní zařízení jednoduše podle šířky HTML dokumentu. Druhou největší modifikací webu je asi menu, které se schovává do strany a je přístupné přes kulaté tlačítko v rohu obrazovky.

3.3. Dvě různá rozložení klávesnice

V České republice jsou běžně používány dvě různá rozložení klávesnice a to QWERTZ a QWERTY. Jedním z problémů, které nevycházely z technického provedení, ale z čistého konceptu, byla implementace těchto dvou rozložení. Nemyslím tím samotnou grafickou klávesnici na frontendu, kde stačilo prohodit Z a Y, ale problém se slovy pro každou lekci zvlášť.

Moje první **nedomyšlené řešení**, které jsem psal ještě v době, kdy stránka disponovala jen texty z náhodně složených písmen, spočívalo v záměně písmenek po vygenerování textu. Tento způsob však logicky fungoval jen pro texty složené z náhodných písmen („ttzz zttz ztzz“) a záměna těchto písmen, ve cvičeních se slovy nebyla ideální („kŭylata yŭstat dŭkay“).

V databázi máme relaci „slovník“, která jak název napovídá, obsahuje všechna slova používaná k trénování. V lekci je ale nutné používat pouze slova složená z **již naučených** písmen. Proto jsem si lekce rozdělil na skupiny podle klávesnice, pro kterou jsou vytvořeny a každému slovu přiřadil id lekce, ve které už je možné ho napsat. Níže v ukázce dat vlevo z databáze můžeme vidět jednotlivé lekce a typ klávesnice, ke kterému patří. Potom v druhé tabulce vpravo vidíme slova ze slovníku a „id“ lekcí, do kterých patří v závislosti na rozložení klávesnice.

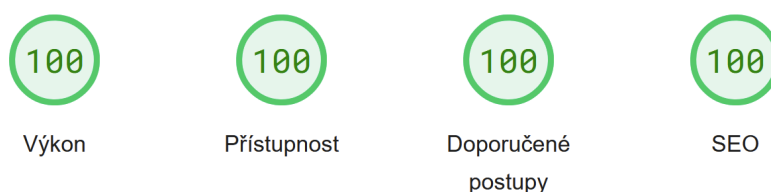
lekce			slovník		
id	písmena	klavesnice	slovo	qwertz_id	qwerty_id
...					
4	aů	oboje			
5	tz	qwertz	autorství	17	17
6	ty	qwerty	autorů	9	9
7	ru	oboje	auty	13	7
...			autory	13	9
12	cn	oboje	autoři	16	16
13	yxm	qwertz			
14	zxm	qwerty			
15	žý	oboje			
...					

Jako příklad se můžeme podívat na slovo auty, které do množiny již použitelných slov nepřichází ve stejné lekci. U varianty QWERTY totiž už umíme všechna potřebná písmena u 7. lekce. Na druhou stranu u QWERTZ, kde u 7. lekce ještě neumíme písmeno Y, se slovo auty objeví až ve 13. lekci, kdy se Y naučíme.

3.4. Optimalizace

SEO (Search Engine Optimization) je něco, čemu by se většina vývojářů raději vyhnula. Bohužel se ale bez toho vaše aplikace k nikomu nedostane. Proto jsem se i na tuto stránku vývoje zaměřil a optimalizoval všechny aspekty aplikace tak, aby byly všechny vyhledávače spokojené a aby hladce běžela na jakkoliv starém hardwaru (testováno na Macbooku, 2008, Intel Core 2 Duo, 4GB RAM, linux).

PageSpeed Insights (běžící na open-source Lighthouse)¹⁰ je **nástroj od Google**, který poskytuje statistiku o výkonu a celkové použitelnosti jakékoliv webové stránky. Nejdůležitější jsou ale doporučení a způsoby, jak veškeré problémy se stránkou vyřešit. I to tento nástroj poskytuje. Poté co jsem prošel každý jeden problém a obětoval pár hodin svého času, jsem dosáhl výsledku níže.



Obrázek 4: PageSpeed Insights screenshot

Trošku nešťastné je **jméno „Jako Pavouk“**, které sice hezky vystihuje aplikaci a umožňuje mi používat pavouka jako maskota, ale překrývá se se stejnojmenným filmem, a tak boj o první příčky ve výsledcích hledání není jednoduchý.

Co se týče samotných vyhledávačů, nejpopulárnější Google má v tuto chvíli paradoxně moji aplikaci rád nejméně. Alternativy jako DuckDuckGo, Qwant nebo Bing zobrazují stránku mnohem častěji a jako jednu z prvních. Naštěstí se ale i pozice na Google vyhledávači postupně zlepšuje.

3.5. Výběr jmen uživatelů

Jelikož Jako Pavouk disponuje i přihlašování pomocí účtů Google, musel jsem se potýkat s otázkou, jaké **unikátní jméno** uživateli přiřadit pro naši databázi, která nemá stejné požadavky na jeho tvar. Od googlu po přihlášení získám kromě e-mailu i jméno a příjmení uživatele, ze kterého mohu vytvořit jméno pro naši aplikaci.

První implementace vypadala tak, že jsem vytáhl jména všech uživatelů z databáze, v lineárním čase je přidal do hashovací tabulky a poté v konstantním čase zkoušel různé kombinace jména a příjmení. Datovou strukturu jsem sice použil, abych se nemusel zbytečně dotazovat do databáze, ale když jsem se podruhé zamyslel, nebylo to nejoptimálnější řešení, jelikož šance, že jméno už bude používáno, je velmi malá. Proto bylo často zbytečné vytvářet hash tabulku jen kvůli jedné jediné kontrole.

Způsob, kterým jsem nakonec první řešení nahradil, spočívá v **přidání náhodného čísla za jméno** uživatele. V tomto případě je velmi malá pravděpodobnost kolize v databázi, a ještě menší, že tato kolize nastane vícekrát. Díky tomuto vylepšení není potřeba načítat všechna jména uživatelů a ani na databázi nebude příliš velký nápor.

4. Design

Vizuální stránka aplikace je podle mého názoru skoro tak důležitá jako její funkčnost. I když staré dobré programy jako 7-Zip nebo FileZilla jsou výjimkou, která potvrzuje pravidlo. Z toho důvodu jsem si dal s vizáží webu práci, a to převážně, co se týče ilustrací.

4.1. Ilustrace

Důležitá pro mě byla jednoduchost a skoro **dětská roztomilost**, aby byl design poutavý pro všechny věkové kategorie. Maskotem se stal **pavouček** jednoduše kvůli jménu aplikace. Pro všechny obrázky, které se objevují na stránce nebo v této dokumentaci, jsem používal vektorový grafický program Adobe Illustrator.

4.2. Barevné schéma

Celá aplikace je na tmavém pozadí a skoro všechn text je bílý. **Světlý motiv** aplikace zatím přidat neplánuji, jelikož by to nejen přidalo na komplexitě, ale jeho používání by bylo nepříjemné například v nočních hodinách.

Fialové odstíny, které jsem si vybral jako analogickou paletu, také nebyly náhodné. Fialová je historicky považovaná za barvu velmi vzácnou, drahou a luxusní. Také údajně působí **uklidňujícím a romantickým dojmem**.¹¹



Obrázek 5: Fialová paleta barev

5. Další vývoj aplikace

5.1. Statistika mezi uživateli

Aby se uživatelé necítily na stránce sami, plánuji přidat různé statistiky, díky kterým by se mohli **porovnat s ostatními**, co se týče rychlosti, přesnosti nebo i prostým procentem dokončených lekcí.

Představuji si hlášky typu: „*Jsi lepší než 62% uživatelů!*“ nebo „*Ještě 12 CPM a budeš mezi stovkou nejrychlejších!*“. U testu psaní by se zase hodil žebříček třeba 100 nejrychlejších pavouků.

5.2. Systém pro školy

Napadají mě dva způsoby jak aplikaci udělat **profitabilní**. Jednou možností je za pár let, až bude aplikace více známá, udělat po vzoru konkurence kurz placený.

Druhá, podle mého názoru lepší varianta, by byla přidat **systém pro učitele**, pomocí kterého by mohli například vytvářet třídy, přidávat žáky, sledovat jejich pokroky a zadávat domácí úkoly.

Tato funkcionality pro školy by byla placená, ať už ve formě roční licence, jednorázové platby, nebo menší částky za každého studenta.

5.3. Věty pro lekce

Trénování na celých větách je zatím možné pouze pro uživatele, kteří si již celou klávesnici osvojili. Důvod je jednoduchý. Nemám k dispozici žádný soubor vět, obsahující pouze **omezený výběr písmen**.

Zkoušel jsem využít sílu, v současné době velmi populárních, LLMs (large language model), ale bohužel moc neporozuměly mému požadavku. Myslím, že tyto modely úplně nerozumí vztahu mezi písmeny a slovy (dotaz 1). Zatím jediným způsobem, jak získat takový soubor, je napsat si věty sám.

Napiš mi 10 vět obsahující pouze písmenka f, j, g, h, d, k, s, l, a, ů, q, ú, w, p, e, o, r, i, t, u, z.

Odpověď Google Gemini:

1. Fialový drak s hnědýma očima klidně seděl na louce.
2. Hasič s houbou uhasí hořící dům.
3. Krásná princezna s úsměvem tančila na plese.

Odpověď ChatGPT:

1. Jak úžasné je, že žluté květy rostou v dále.
2. Vůně jasmínu zdobí chodbu hotelu.
3. Klidně si odpočiň pod stromem.

Dotaz 1: (březen 2024)^{12,13}

6. Závěr

Výsledkem této práce je webová aplikace, která nejen splňuje stanovené požadavky, ale také přesahuje mé očekávání.

Naučil jsem se všechno okolo vývoje backendu. Od napojení na databázi až po ošetření nejrůznějších bezpečnostních skulinek. Také jsem se zorientoval v konfiguraci linuxových serverů na produkci. V tom mi cestu velmi usnadnil studentský vývojový server a jeho pohledný, smělý a velmi inteligentní admin. Díky tomu, že mi bylo svoleno k použití moderního jazyku Go, jsem si ho nejen mohl porovnat s jazyky, ve kterých jsem si již byl jistější, ale také mě přiblížil k více low level programování. S Vue jsem už sice zkušenosti měl, ale znalosti týkající se SEO určitě v budoucnu využiji.

Myslím, že ve vývoji aplikace budu dále pokračovat, jelikož mě baví vyvíjet něco, na co se může moje mamka podívat a být na mě pyšná, že už na počítači jen nehraji hry.

Odkazy

- 1 Vue.js logo. https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Vue.js_Logo_2.svg/1200px-Vue.js_Logo_2.svg.png
- 2 PostgreSQL logo. https://upload.wikimedia.org/wikipedia/commons/thumb/2/29/Postgresql_elephant.svg/993px-Postgresql_elephant.svg.png
- 3 Go logo. https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Go_Logo_Blue.svg/1200px-Go_Logo_Blue.svg.png
- 4 Repozitář jazyku Typst. <https://github.com/typst>
- 5 Historie rozložení kláves. https://cs.wikipedia.org/wiki/Rozlo%C5%BEen%C3%AD_kl%C3%A1ves
- 6 QWERTZ rozložení. https://cs.wikipedia.org/wiki/Psac%C3%AD_stroj
- 7 Psací stroj. https://as1.ftcdn.net/v2/jpg/06/34/81/28/1000_F_634812812_h5mTXqkvNsiB2mkz1uLoR6O1QBi7x4L.jpg
- 8 Pohádkozem. <https://www.pohadkozem.cz/>
- 9 Mobile vs Desktop statistika. <https://www.semrush.com/blog/mobile-vs-desktop-usage/>
- 10 Lighthouse. <https://developer.chrome.com/docs/lighthouse/overview/#psi>
- 11 Teorie barev, Fialová. <https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/#purple-secondary-color>
- 12 Google Gemini. <https://gemini.google.com/>
- 13 ChatGPT. <https://chat.openai.com/>