

Gymnázium, Praha 6, Arabská 14

Obor Programování

Ročníková práce

Opakuj!



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

Miroslav Sklenář

Anotace

Úkolem je vytvořit aplikaci pro opakování maturitních otázek (otázky a odpovědi by vytvářel sám uživatel podle vzorového textového souboru). Za každou správnou odpověď by jakoby zaútočil na nepřítele a tím mu ubíral životy (počet otázek k danému maturitnímu tématu), pro hráče by platilo, že za špatnou odpověď by mizely životy jemu (měl by omezený počet např. 3). Způsob otázek by si mohl také vybrat např.. spojovačka, výběr správné odpovědi, napsat odpověď,... Program bude implementovat učící algoritmus, který bude vybírat otázky k zodpovězení na základě správnosti předchozích odpovědí.

Obsah

1	Úvod	3
1.1	Myšlenka za aplikací	3
2	Front-end webové stránky	4
2.1	Domovská stránka	4
2.2	Rozložení "navbaru"	4
2.3	Stránka "Prohlízet"	4
2.4	Stránka "Pomoc"	5
2.5	Stránka "Má cvičení"	5
2.6	Stránka "Vytvořit cvičení"	5
2.7	Stránka s profilem uživatele	6
2.8	Skripty na front-endu	6
2.8.1	ExerciseScript.ts	6
2.9	Vuex.js	7
2.10	Vuetify.js	7
3	Back-end webové stránky	8
3.1	Server	8
3.1.1	Express.js	8
3.1.2	Databáze	8
3.1.3	Datábázová schémata	8
	User.ts	8
	ExercisePrivate.ts a ExercisePublic.ts	9
	UserStatistics.ts	10
3.1.4	Vue.js	10

3.2	Přihlášení	11
3.2.1	Skript Auth.js	11
3.3	Serverové skripty	11
3.3.1	ExercisesCheck.ts	11
3.3.2	Exercises.ts	11
4	Závěr	12
	Zdroje	13
	Seznam příloh	14

1 Úvod

Na úvod bych chtěl říci, že aplikace je dělána pro studenty, kteří potřebují opakovat - procvičovat než-li učit. Aplikace měla být takto koncipována už od začátku, ale nedostatek nadhledu autora, způsobil, že aplikace se v některých částech odloučila od původního znění zadání, ale duch této aplikace zůstal stejný, a to umožnit studentům zopakovat si již vypracované témata/otázky o trochu zajímavějším způsobem než je jen slepě recitovat z popsaného papíru.

1.1. Myšlenka za aplikací

Prvotní myšlenkou za touto aplikací stál pocit nutnosti mít nějakou pomůcku k učení/opakování na maturitu. Bohužel, ale tato myšlenka přišla příliš pozdě, ideální by bylo, kdyby přišla o rok či dva dříve, aby byl čas ji doladit, popřípadě dodělat včas, kdy by byla ještě k užitku. No na začátku školního roku, to připadalo autorovi jako celkem dobrý nápad, který by mohl mít i dost užitku, ten ale nepředpokládal, že vývoj by mohl zabrat tolik času.

2 Front-end webové stránky

2.1. Domovská stránka

V domovské stránce - to je stránka, která se uživateli zobrazí po prvotním načtení webové stránky, se nachází pouze uvítání uživatele. Uživatel zde nemůže nic dělat, stránka je zde pouze s estetického hlediska, a tak musí uživatel na části stránky označované jako *navbar* se buď přihlásit nebo jít na stránku s veřejnými cvičeními (pojmenované "Prohlížet") či na stránku s návody ("Pomoc").

2.2. Rozložení "navbaru"

Navbar je označení pro horní lištu na stránce, nejčastěji s tlačítky pro ostatní stránky. V této aplikaci je navbar rozdělen na dvě poloviny. V levé polovině se nachází odkazy pro předměty stránky - cvičení, nápověda, domovská stránka,... a pravá polovina, kde se nachází odkazy týkající se uživatele a to přihlášení, odhlášení, profil uživatele a také tlačítko pro změnu barevného tématu stránky - tmavé a světlé, výchozí nastavení tématu vychází ze systémového tématu.

2.3. Stránka "Prohlížet"

Na této stránce se nachází cvičení, která byla označena jako *Public*, což znamená veřejné, a proto jsou přístupná i nepřihlášeným uživatelům. Pokud však chce někdo přidat cvičení mezi veřejná, tak musí být přihlášen a také cvičení mít napsané v textovém formátu (.txt) podle instrukcí ze stránky "Pomoc". Poté jej může přidat po kliknutí na zelené tlačítko se symbolem plus na stránce "Má cvičení".

2.4. Stránka "Pomoc"

Zde se nachází návody jak napsat cvičení, jak je nahrát na server (do databáze), jaké mají mít formy, atd.. Tato stránka slouží pouze k informování uživatele, a ten tak zde nemůže nic dělat.

2.5. Stránka "Má cvičení"

Na stránce "Má cvičení" se nachází seznam všech cvičení, která uživatel kdy nahrál. Tato cvičení jsou reprezentována "kartičkami", kdy každé cvičení má tvar oblého čtverce a pozadí tohoto čtverce má odlišnou barvu od pozadí stránky. Na této "kartě" se nachází tři tlačítka - pro spuštění, pro úpravu a pro odstranění. Pokud by uživatel chtěl odstranit cvičení, tak je po stisknutí tlačítka dotázán ještě jednou, zda si opravdu přeje smazat cvičení. Cvičení jsou řazena do řad a tato řada je vždy zakončena zeleným tlačítkem se znaménkem plus. Toto tlačítko se také nachází na vrchu stránky, pro případ, že by uživatel měl příliš mnoho cvičení a tak by nechtěl hledat zelené tlačítko někde na spodku stránky. Bohužel upravování cvičení, nefunguje, jak by mělo, vzhledem k tomu, že během vývoje se zjistilo, že přetěžuje prohlížeč a proto není doporučováno jeho používání, i když se dá říct, že "funguje". Zelené tlačítko s pootočeným trojúhelníčkem přesune uživatel na stránku, kde si může přizpůsobit obtížnost daného opakování - ta je rozdělena na tři úrovně - lehká, střední a obtížná

2.6. Stránka "Vytvořit cvičení"

Stránka vytvořit cvičení, je, jak název napovídá určena k vytváření cvičení. To je zajištěno formátem formuláře, kdy jsou deklarována povinná pole - jméno tématu (*name*), jméno otázky (*question*) a typ otázky (*type*), ten je vybírán pomocí HTML funkce *select*, kdy uživatel má na výběr z předem definovaných hodnot. Po vybrání typu otázky, se uživateli zobrazí adekvátní rozhraní pro danou otázku - pro dopňování do věty se zobrazí pole pro větu a pole pro odpovědi, pro jednu odpověď na otázku, se zobrazí jen pole pro odpověď, pro dvojice se vygenerují dvě pole pro text, pro vybírání správných odpovědí mezi nesprávnými, se zobrazí dva sloupce, každý pro

vlastní skupinu odpovědí (správné, špatné). U otázek, kde je možné zadat více správných odpovědí, je vždy tlačítko se znaménkem plus pro přidání dalšího pole pro danou odpověď. Pokud je cvičení hotové - tzn. že podmínka pro formulář je splněna - všechny požadované hodnoty jsou vyplněny, tak uživatel může finalizovat svou tvorbu kliknutím na tlačítko v pravém horním rohu rozhraní vytvářeče cvičení s nápisem "Hotovo", kdy odešle objekt *exercise* serveru, který jej zkontroluje a pokud je vše jak má být, tak pošle pozitivní odpověď, která pak uživatele přesměruje na stránku "Má cvičení".

2.7. Stránka s profilem uživatele

Zde se uživatel ocitne po přihlášení, tím se může ujistit, že je opravdu přihlášený. Také zde může vidět své statistiky, pokud již nějaká cvičení odehrál, zpracované do koláčového grafu pomocí knihovny *Chart.js*.

2.8. Skripty na front-endu

2.8.1. ExerciseScript.ts

V tomto skriptu je převáděn textový soubor na objekt *Exercise* podle předem stanovených norem. Každý soubor se cvičením musí mít na prvním řádku *#JménoTématu* následované na dalším řádku otázkou, která musí začínat číslem 1, poté tečkou a následně otázkou, poté musí na dalším řádku být *Typ*- a značka pro daný typ otázky - \$, %, &, * - doplnění do textu, jedna správná odpověď, spojování dvojic, výběr správných odpovědí. Následně musí být další otázka oddělena prázdnou řádkou, ale uživatel si musí zároveň dát pozor, aby soubor nekončil prázdnou řádkou, jinak mu aplikace vyhodí error, když bude chtít cvičení nahrát.

2.9. Vuex.js

Vuex umožňuje ukládat data mimo prostředí *Vue*, a proto je vhodné pro uložení např. uživatelského profilu či uživatelské session, kdy tyto data jsou přístupná kdykoliv a kdekoliv na stránce.

2.10. Vuetify.js

Vuetify.js je komponentový framework pro Vue.js. Tento framework využívá systém komponentů z frameworku Vue.js, kdy má v sobě "knihovnu" předem nadesignovaných komponent připravených k použití. Tyto komponenty často přichází i s řadou skriptů pro manipulaci či kontrolu údajů. Proto je velice snadné složit relativně komplexní stránky s patřičnou funkcionalitou. Bohužel tento framework je opravdu zamýšlen jakožto knihovna prefabrikovaných komponent, a tak neumožňuje jemnou manipulaci s prvky. A proto bylo Vuetify doplněno o framework *Tailwind CSS*, který právě umožňuje jemnou konfiguraci vybraných prvků.

3 Back-end webové stránky

3.1. Server

3.1.1. Express.js

Server je vytvářen skrze framework *Express.js* pro *Node.js*, jsou v něm nastavovány funkce, které se mají spustit společně se serverem. Např. připojení knihovny *Passport.js* či inicializování routeru pro login a API (cesty routeru pro komunikaci mezi front-endem a back-endem).

3.1.2. Databáze

Databáze je zajištěna cloudově přes *MongoDB*. Komunikaci serveru a databáze zajišťuje knihovna *mongoose*. *Mongoose* umožňuje vytváření schémat do databáze, zjednodušuje komunikaci a také zajišťuje bezpečné manipulování s databází. To všechno byly důvody, proč jsem jej zvolil pro mou aplikaci.

3.1.3. Datábázová schémata

User.ts

Toto je schéma pro uživatele, které je vyplněno při přihlášení uživatele pomocí GSI. Jediný údaj, který je vygenerován lokálně je `userID`. `UserID` je vygenerováno pomocí knihovny *NanoID*, ta zajišťuje nenáročně a zároveň bezpečné generování ID. Pod tímto odstavcem lze vidět zjednodušené zápis schématu pro *mongoose*.

```

1  googleID: String;
2  displayName: String;
3  firstName: String;
4  lastName: String;
5  email: String;
6  pfp: String;
7  userID: String;

```

Příloha 3.1: [Kód]Schéma pro mongoose pro uživatele

ExercisePrivate.ts a ExercisePublic.ts

V tomto schématu se nachází zápis cvičení pro soukromé účely - *ExercisePrivate* a zároveň pro *ExercisePublic*. Tato schémata jsou stejná, ale zároveň jsou oddělena pro jednodušší používání a liší se jen v hodnotě *visibility*. Pod tímto odstavcem lze vidět zjednodušené zápis schématu pro mongoose.

```

1  userID: String;
2  name: String;
3  question: [{
4      number: number;
5      question: String;
6      type: String;
7      context: String;
8      answer: String;
9      answersRight: [[{ String }]];
10     answersWrong: [[{ String }]];
11 }]];
12 exerciseID: String;
13 visibility: String;

```

Příloha 3.2: [Kód]Schéma pro mongoose pro cvičení

UserStatistics.ts

Zde se ukládají statistiky uživatele k zobrazení na jeho profilu. Ne všechny atributy jsou využity přímo na stránce profilu, ale jsou k dispozici v databázi pro budoucí využití. Do databáze je ukládán počet všech odehraných cvičení, počet všech odpovědí, počet všech správných i špatných odpovědí, počet bezchybně dokončených cvičení, ale také počet nezvládnutých cvičení a nakonec i pole identifikátorů (ID) všech bezchybných nebo naopak nezvládnutých cvičení, pro případné zobrazování na stránce profilu.

```
1  userID: String;
2  totalExercisesPlayed: number;
3  totalAnswers: number;
4  totalCorrectAnswers: number;
5  totalIncorrectAnswers: number;
6  totalAcedExercises: number;
7  totalFailedExercises: number;
8  exactFailedExercises: [{ String }];
9  exactAcedExercises: [{ String }];
```

Příloha 3.3: [Kód]Schéma pro mongoose pro statistiky uživatele

3.1.4. Vue.js

Pro generování front-endu bylo použito Vue.js. To umožňuje rozdělit složitá uživatelská rozhraní do jednodušších *component*, které je možné využít znovu a zároveň jinde. Také je celý klient generován client-side, to znamená, že HTML rozhraní, které uživatel vidí je vytvořeno přímo u něj v prohlížeči. Protože, ale server musí posílat spoustu dat klientovi, tak je příliš náročný pro free hosting a tak nemůže být nikde hostován zadarmo a proto také není přístupný na internetu. Mimo to také byl použit framework *VuetifyJS* pro zjednodušení návrhu a designu jednotlivých stránek. Zároveň byl ale využit framework kaskádových stylů *Tailwind.css* pro doladění jednotlivých prvků.

3.2. Přihlášení

Přihlášení je zajištěno skrze Gmail a GSI(Google sign-in) a není tedy nutné zajišťovat registrace nových uživatelů. Přihlášení je zpracováváno skrze knihovnu google-one-tap middlewaru *Passport.js*. Uživatelský profil je vytvořen do databáze s poskytnutými daty od Googlu popřípadě pokud již existuje záznam o uživateli v databázi, tak jsou jeho data načtena odtud.

3.2.1. Skript Auth.js

V *auth.js* je spuštěna funkce *use* middlewaru *Passport.js* a té se přiřadí objekt *Strategy* z knihovny *google-one-tap*. Ta se postará o ověření a zpracování přihlášení uživatele, který se přihlašuje skrze Gmail, pokud uživatel již není v databázi, tak vytvoří jeho záznam. Pokud již uživatel existuje, tak jej pustí dál.

3.3. Serverové skripty

3.3.1. ExercisesCheck.ts

Tento soubor obsahuje několik funkcí, které mají kontrolovat, zda nenastal v objektu *Exercise* nějaký nečekaný status a to zejména undefined nebo null. Tyto dvě hodnoty by způsobily problémy ve front-endové části aplikace a tak se musí zajistit, že se nikdy nedostanou do databáze. To je zajištěno několika funkcemi, které kontrolují podmínky pro různé atributy z objektu *exercise* nebo také podobjektu *question*.

3.3.2. Exercises.ts

Zde nachází funkce *checkJSON* a v ní se použije v cyklu funkce z *ExercisesCheck.ts* na objekt *exercise* a pokud něco není v pořádku, tak změní boolean stav proměnné *isItOk*, který tato funkce vrací. Tento soubor obsahuje také dvě funkce pro uložení cvičení do databáze a to pro veřejnost - public a soukromě - private.

4 Závěr

Práce si myslím, že se povedla, samozřejmě vždy je prostor pro zlepšení a to platí i pro moji aplikaci. Je škoda, že aplikace je příliš náročná na free hosting, jinak by bylo skvělé ji mít někde běžet a tedy i využívat. Myslím si, že práce splňuje základní nároky studenta, který si chce opakovat již zpracované otázky/témata trochu zajímavěji než prostým přeříkáváním vypsanych informací, ale je zde prostor pro "opakovací" prvky, a to zejména onen učící algoritmus. Ten je inspirován chováním aplikace Duolingo, avšak je mnohem "hloupější" a tudíž má obrovské prostory pro zlepšení a zpestření učícího procesu, ale věřím, že většině studentů by stačil. Kdybych měl celou práci shrnout, tak si myslím, že byla pro mě velice přínosná, protože jsem se naučil pracovat s TypeScriptem a alespoň trochu jsem se zorientoval v prostředí frameworku Vue.js.

Zdroje

- [1] *Axios get request error*. URL: <https://stackoverflow.com/questions/60076226/html-in-vue-js-not-updating-after-axios-get-request-updates-objects-property> (Navštíveno dne 02/16/2024).
- [2] *Google-one-tap*. URL: <https://stackoverflow.com/questions/tagged/google-one-tap?tab=Frequent> (Navštíveno dne 11/07/2023).
- [3] *PrimeVue Chart*. URL: <https://primevue.org/> (Navštíveno dne 03/29/2024).
- [4] *Problém s načítáním CSS a JS souborů*. URL: <https://stackoverflow.com/questions/63007293/vuejs-loading-css-and-js-files-giving-mime-type-error> (Navštíveno dne 01/07/2024).
- [5] *Tailwind CSS*. URL: <https://tailwindcss.com/> (Navštíveno dne 02/20/2023).
- [6] *Vite.js*. URL: <https://vitejs.dev/> (Navštíveno dne 11/07/2023).
- [7] *Vue*. URL: <https://vuejs.org/> (Navštíveno dne 11/07/2023).
- [8] *Vue Guide*. URL: <https://vuejs.org/tutorial/#step-1> (Navštíveno dne 11/07/2023).
- [9] *Vue&Vite&Express*. URL: <https://github.com/szymmis/vite-express> (Navštíveno dne 11/07/2023).
- [10] *Vuetify*. URL: <https://vuetifyjs.com/en/> (Navštíveno dne 12/23/2023).
- [11] *Vuetify change event in v-select*. URL: <https://stackoverflow.com/questions/71081377/vuetify-3-change-event-in-v-select> (Navštíveno dne 02/19/2024).
- [12] *Vuetify Guide*. URL: <https://vuetifyjs.com/en/components/all/#containment> (Navštíveno dne 12/23/2023).

Seznam příloh

3.1	[Kód]Schéma pro mongoose pro uživatele	9
3.2	[Kód]Schéma pro mongoose pro cvičení	9
3.3	[Kód]Schéma pro mongoose pro statistiky uživatele	10