

GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Webová aplikace na anonymní zpětnou vazbu z výuky

Maturitní práce

Autor: Matyáš Neruda

Třída: 4.A

Školní rok: 2020/2021

Předmět: Informatika

Vedoucí práce: Šimon Schierreich

Praha, 3.3.2024



GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student: Matyáš Neruda
Třída: 4.A
Školní rok: 2021/2022
Platnost zadání: 30. 9. 2022
Vedoucí práce: Šimon Schierreich

Název práce: Aplikace na hodnocení výuky

Pokyny pro vypracování:

Cílem práce je vytvořit webovou aplikaci, která bude vyučujícím a vedení školy umožňovat definici otázek, na něž chce od studentů získat zpětnou vazbu. Tyto otázky se mohou pro jednotlivé předměty lišit. Žáci budou naopak mít možnost ke studovaným předmětům anonymně přidávat odpovědi na zadané otázky. Aplikace bude dále umožňovat procházení anketních otázek a studentských odpovědí z minulých let.

Doporučená literatura:

- [2] EVANS, Eric. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison Wesley Longman Publishing, 2003. ISBN 978-0-321-12521-7.
- [3] MARTIN, Robert C. *Design Principles and Design Patterns*. www.objectmentor.com, 2000. Dostupné z: https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf.
- [4] FOWLER, Martin. *Patterns of enterprise application architecture*. Boston: Addison-Wesley Professional, 2003. ISBN 978-0-321-12742-6.

URL repozitáře:

<https://github.com/KingMaty/maturitni-projekt.git>

student

vedoucí práce

V Praze dne 3. 3. 2024

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 4. března 2024

Matyáš Neruda

Poděkování

Vážení rodiče. Chtěl bych vám touto cestou vyjádřit upřímné poděkování za veškerou podporu, kterou jste mi poskytovali během mého studia. Vaše trpělivost, povzbuzování a neustálá motivace hrály klíčovou roli v mém vzdělávacím procesu a já si toho velice vážím. Panu učiteli Schierreichovi děkuji za jeho čas a dobré rady při konzultaci maturitní práce.

Abstrakt

Maturitní práce se zaměřuje na vývoj webové aplikace na anonymní zpětnou vazbu z výuky. V první, teoretické části, je podrobně popsán význam funkcí aplikace pro jednotlivé role uživatelů. Druhá část se věnuje popisu backendu aplikace. Zde je popsána manipulace s daty v Pythonu s využitím frameworku Flask. Jsou zde detailně popsány použité routy a dotazy potřebné pro práci s databází, stejně jako návrh tabulek v databázi. Ve frontendové části práce je popsána tvorba základní struktury stránek pomocí HTML a šablonovacího jazyka Jinja. Dále je zde popsána implementace interaktivních prvků a stylování pomocí JavaScriptu, CSS a frameworku Bootstrap.

Klíčová slova

HTML, SQL, Python, Flask, Bootstrap, Jinja, Web, Aplikace, Student, Učitel

Abstract

The graduation thesis focuses on the development of a web application to anonymous feedback from teaching. The first, theoretical part, details the importance of the functions of the application for individual user roles. The second part is devoted to describing the backend of the application. This describes the manipulation of data in Python using the Flask framework. There are detailed descriptions of the used routes and queries needed to work with the database, as well as the design of the tables in the database. The frontend part of the work describes the creation of the basic page structure using HTML and the Jinja template language. It also describes the implementation of interactive elements and styling using JavaScript, CSS and the Bootstrap framework.

Keywords

HTML, SQL, Python, Flask, Bootstrap, Jinja, Web, Application, Student, Teacher

Obsah

1	Teoretická část	3
1.1	Funkce pro uživatele	3
1.1.1	Funkce pro učitele	3
1.1.2	Funkce pro studenty	3
2	FrontEnd	5
2.1	Implementace frontendu webové aplikace	5
2.2	HTML	5
2.3	Bootstrap	6
2.4	Jinja	7
3	BackEnd	9
3.1	Framework Python Flask	9
3.2	MySQL	9
3.3	Implementace databáze	9
3.4	Autentizace	9
3.5	Tabulky	10
3.5.1	Nezávislé:	10
3.5.2	Data zpětné vazby:	10
3.5.3	Relace:	10
	Závěr	13
4	Citace	15
4.1	15
4.2	15
4.3	15
4.4	15
	Seznam obrázků	17
	Seznam tabulek	18

1. Teoretická část

Tato maturitní práce se zabývá řešením problému získávání zpětné vazby od studentů ve výukovém procesu. Cílem tohoto projektu je vytvořit webovou aplikaci, která umožní vyučujícím získávat zpětnou vazbu od svých studentů prostřednictvím otázek, které si sami definují. Cílem je zlepšení výuky a prostředí ve škole pro obě strany - učitele i studenty.

1.1 Funkce pro uživatele

Aplikace poskytuje užitečný nástroj pro snadnou definici otázek, sběr odpovědí a analýzu dat z minulých let. Mohou také procházet historií odpovědí na otázky napříč roky, což pomůže učitelům sledovat vývoj jejich vyučovacích metod. Studentům na druhou stranu zkušenosti starších spolužáků budou moci sloužit jako pomůcka při výběru volitelných předmětů.

1.1.1 Funkce pro učitele

Učitelé mají možnost definovat specifické otázky pro jednotlivé předměty, což je klíčové, protože obsah a požadavky se mohou mezi jejich předměty lišit. Dále mohou přidávat nové otázky do systému prostřednictvím formuláře a zobrazovat seznam otázek svých aktivních otázek, na které mohou studenti odpovídat. Nebude chybět ani možnost špatně položené, nebo jen nechtěné otázky smazat ještě předtím, než na ně studenti stihnou zareagovat.

1.1.2 Funkce pro studenty

Studenti mají možnost odpovídat na otázky prostřednictvím formuláře. Mohou také procházet historií odpovědí na otázky napříč různými roky a využívat to k efektivnímu zlepšování svého studia. Dále mohou zobrazovat seznam aktivních otázek, na které mohou v danou chvíli odpovídat. Aplikace dále poskytuje přehled otázek a odpovědí z minulých let, což mimo jiné umožňuje sledovat zkušenosti starších spolužáků s jednotlivými vyučujícími a volitelnými předměty. Studenti na druhé straně budou mít možnost anonymně odpovídat na otázky, které se jich týkají.

2. FrontEnd

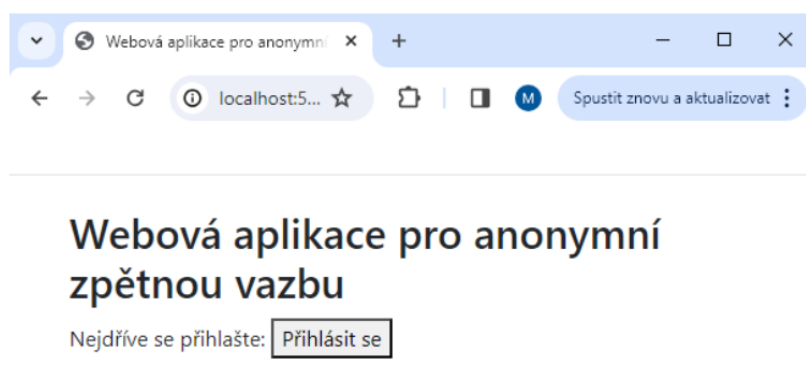
2.1 Implementace frontendu webové aplikace

Pro implementaci frontendu naší webové aplikace byly použity technologie a knihovny, které mi umožnily vytvořit responzivní a interaktivní uživatelské rozhraní.

2.2 HTML

Implementace HTML souborů na konkrétních příkladech:

Základní šablona (base.html): Tento soubor slouží jako základní šablona pro všechny stránky v naší aplikaci, s výjimkou pro login. Díky tomu jsem nemusel do každého souboru zvlášť vkládat kódy potřebné pro menu, stačilo na začátek každé šablony napsat Soubor login.html slouží pro přihlášení se do webové aplikace pro studenty a učitele.

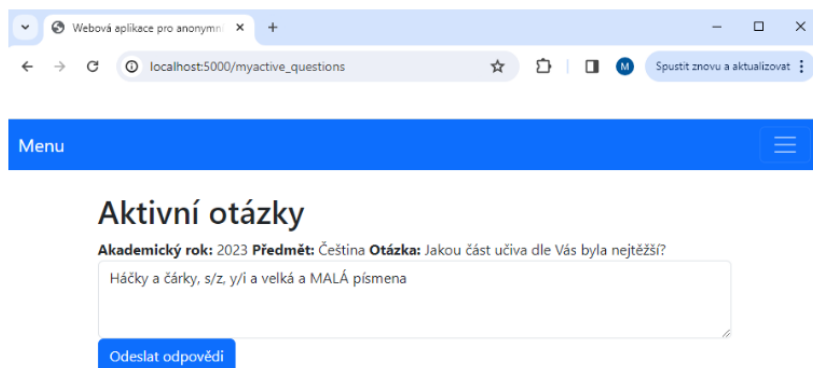


Obrázek 2.1: login.html

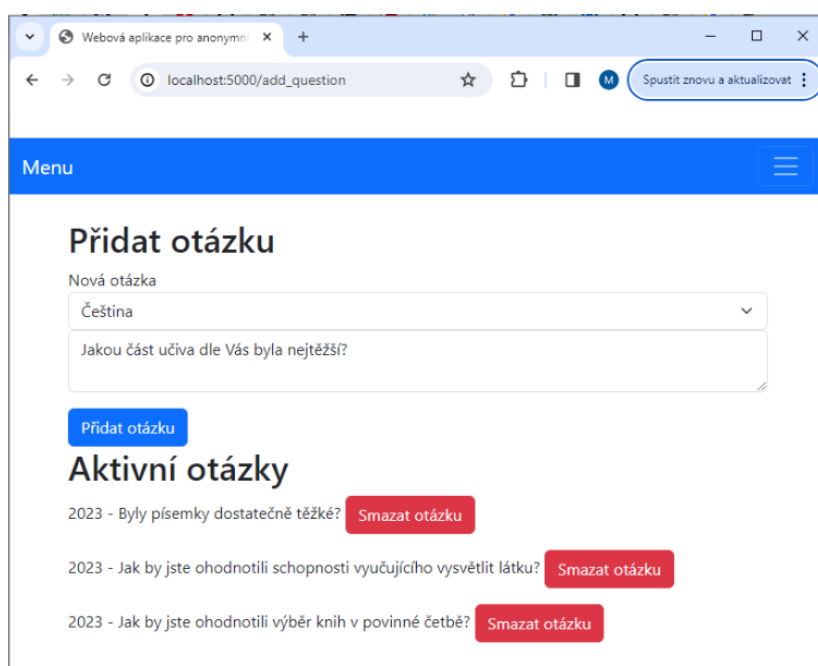
Aktivní otázky (myactive_questions.html): Tento soubor je určen k zobrazení seznamu otázek, na které mohou studenti v danou chvíli odpovídat. Díky našemu backendu se zde každému studentovi zobrazují pouze otázky, které se ho týkají. U každé otázky jsou vypsané všechny informace, které student potřebuje znát, aby věděl na co přesně odpovídá. K jejich vypsaní jsem využil: `<label for="answer question[o]">Akademický rok: question[3] Předmět: question[4] Vyučující: question[o] Otázka question[1]</label>`. Obsahuje formulář pro přidávání odpovědí a seznam zodpověditelných otázek. Formulář skládající se z libovolného počtu otázek, končí jediným tlačítkem, které uživateli umožní odpovědět na všechny otázky najednou.

Přidání otázek (add_question.html): Tento soubor slouží k přidávání nových otázek. Obsahuje dropdown komponentu pro výběr předmětu, ke kterému chce učitel zrovna přidat otázku a pole pro zadání nové otázky. Jednotliví vyučující zde také vidí seznam jimi přidávaných otázek s tlačítkem smazat u každé z nich, aby se nikomu omylem nepodařilo smazat jinou otázku než zamýšlel.

Historie odpovědí (myanswers.html): Tento soubor slouží k zobrazení historie odpovědí uživatelů na otázky napříč rokama. Umožňuje jak učitelům, tak studentům procházet všechny otázky sloužící



Obrázek 2.2: myactivequestions.html



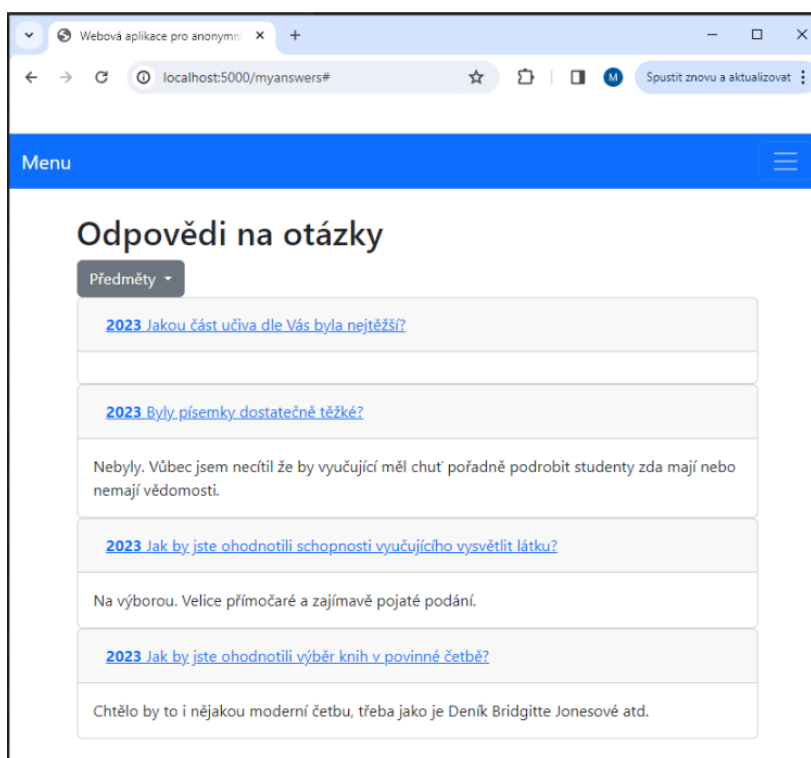
Obrázek 2.3: addquestion.html

jako linky pro zobrazení odpovědí. Ty jsou i tady seřazeny od nejnovějších po nejstarší. Obsahuje dropdown link pro výběr předmětu, ke kterému hledáte zpětnou vazbu. Třídění je úmyslně nezávislé na učitelích, to umožní zobrazit více otázek najednou, jejichž odpovědi pak můžeme

Soubor MyDbConnError.html je HTML stránka, která se zobrazí pouze pokud je problém s připojením do databáze.

2.3 Bootstrap

Integrace frameworku Bootstrap mi umožnila využít širokou škálu předdefinovaných komponent, jako jsou tlačítka, navigační panely, formulářové prvky a mnoho dalšího. Poskytl také responzivní mřížový systém, který umožnil naší aplikaci automaticky přizpůsobit se různým zařízením. Tyto komponenty jsem díky přehledné dokumentaci na stránkách vývojářů snadno integroval rovnou



Obrázek 2.4: myanswers.html

do našich HTML šablon pomocí jednoduchých kódů např.: `<nav class="navbar navbar-expand-lg navbar-dark bg-primary">` což je element definující interaktivní hlavičku a její responzivní rozměry. Jednotlivé navigační položky se pak definují třídou (`<li class="nav-item">`). Pak už jen stačilo do hlavní souboru `mybase.html` přidat vývojářem poskytnuté odkazy.

2.4 Jinja

Jinja je šablonovací systém pro jazyk Python, který umožňuje vkládat dynamický obsah do statických šablon. Tento systém používán k generování dynamických HTML stránek. Jinja umožňuje vkládat proměnné, podmínky, smyčky a další konstrukce přímo do HTML šablon, což umožňuje tvorbu šablonových souborů s proměnlivým obsahem. Jinja šablony jsou vykreslovány serverem při požadavku na danou URL adresu. To umožňuje dynamicky generovat obsah stránky.

3. BackEnd

3.1 Framework Python Flask

Jako základ jsem zvolil Python Flask. Jedná se o webový framework napsaný v jazyce Python, který díky svým knihoven nabízí flexibilitu a snadnou použitelnost. Je známý pro svou jednoduchost a existuje na něj spousta tutoriálů. Jednotlivé knihovny se importují pomocí příkazu `import` a názvu modulu/knihovny `mysql,connector, requests...`. V případě přidání vlastního modulu je potřeba nutno doplnit i adresu odkud přidáváme jako jsou blueprinty například takto: `from auth import auth_bp`.

3.2 MySQL

Jako relační databázový systém pro správu dat používá aplikace MySQL. Byl zvolen protože jej lze využívat zdarma s licencí GPL a dá se velmi snadno nainstalovat a zprovoznit. Zároveň je to jeden z nejpoužívanějších databázových systémů, díky čemuž obsahuje rozsáhlé dokumentace. Tento bod se zabývá návrhem a implementací databáze, včetně definice tabulek, vztahů mezi nimi a manipulace s daty. Struktura databáze obsahuje tabulky pro uchování informací o uživateli, předmětech, otázkách a odpovědích. Propojení těchto tabulek pomocí cizích klíčů nám umožňuje implementovat do aplikace jednotlivé funkce potřebné pro získávání zpětné vazby. Prvním krokem pro implementaci databáze byl návrh tabulek.

3.3 Implementace databáze

Pro propojení webové aplikace s databází MySQL jsem využil knihovnu `mysql.connector`. Tato knihovna mi umožnila vytvořit spojení s databází pomocí konfiguračních údajů `{'host', 'user', 'password' a 'database'}`, které jsem uložil do proměnné `DB_CONFIG`. Poté jsem vytvořil funkci `connect()`, která se pokusí vytvořit spojení s databází pomocí `mysql.connector.connect(**DB_CONFIG)`. Pokud se spojení vytvoří úspěšně, funkce vrátí objekt spojení. Pokud dojde k chybě, funkce vrátí `None`. Dále jsem vytvořil funkce `insert_data()` a `delete_data()`, které umožňují vkládání a mazání dat z databáze. Tyto funkce přijímají objekt spojení, SQL dotaz a data pro dotaz jako parametry. Využívají metodu `execute()` objektu `cursor` pro provedení SQL dotazu a metodu `commit()` objektu spojení pro uložení změn do databáze. Konečně, jsem vytvořil funkci `insert_answer_and_student_question`, která využívá funkci `insert_data()` pro vložení odpovědi a otázky studenta do databáze.

3.4 Autentizace

Pro autentizaci uživatelů jsem měl použít Google OAuth. Toto rozhodnutí zajišťuje bezpečnost a pohodlí uživatelů, kteří se tak mohou přihlásit pomocí svého stávajícího Google účtu. Pro komunikaci s Google API a získání autorizačních toků jsem využil modul `Requests`. Nejprve jsem získal

autentizační údaje pomocí `flow.credentials` a následně vytvořil session pro HTTP požadavky pomocí `requests.session()`. Tato session byla uložena do mezipaměti pomocí `cachecontrol`, což mi umožnilo omezit opakované požadavky na server. Pro integraci OAuth autentizace byla použita knihovna `Flask-OAuthlib`. Po vytvoření projektu v Google Developers Console a získání autentizačních klíčů jsem definoval routy pro přihlášení a odhlášení uživatelů pomocí metod poskytovaných rozšířením `Flask-OAuthlib`.

3.5 Tabulky

Těch bylo potřeba osm. Musel jsem totiž zařídit, aby mohl každý učitel přidávat otázky pouze k předmětům, které učí a pouze v roce, kdy je učí. A aby každý student mohl zas odpovídat jen na otázky vztahující se ke konkrétním předmětům, které se učí. Pro snazší pochopení se dají rozdělit do tří skupin:

3.5.1 Nezávislé:

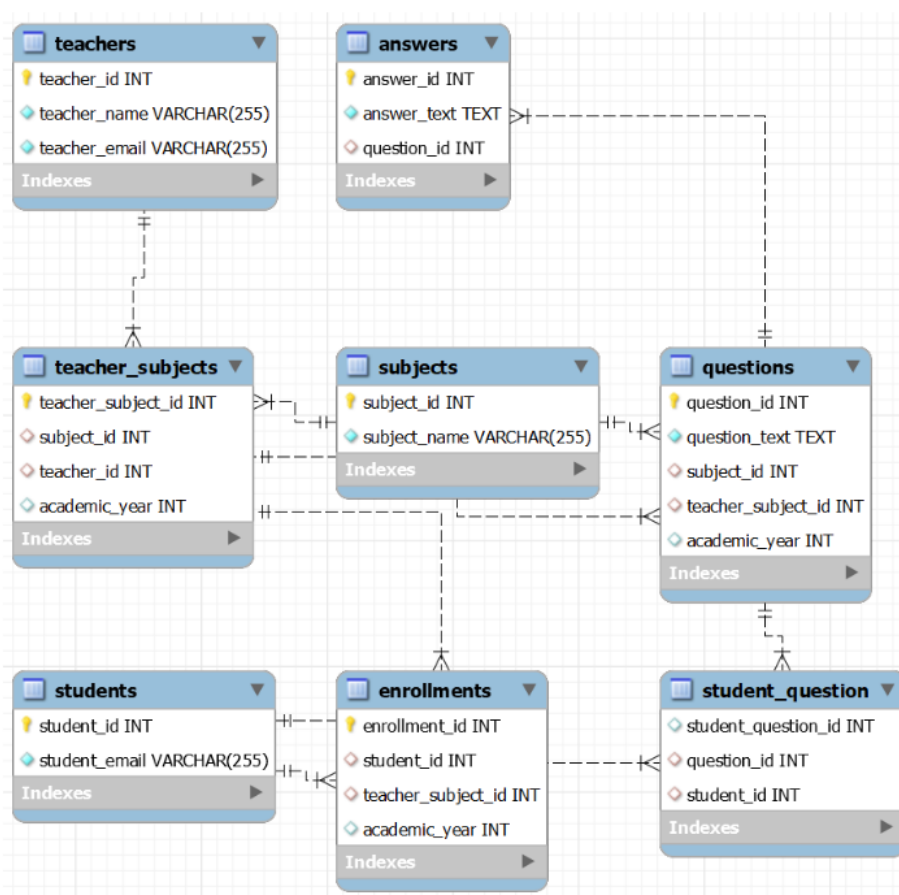
Do této kategorie patří tabulky `students`, `teachers` a `subjects`, nesoucí hlavní klíče o uživatelích a předmětech.

3.5.2 Data zpětné vazby:

Zde jsou tabulky `questions` a `answers`. Ty slouží pro správu uživateli přidaných dat jako jsou otázky, odpovědi a rok jejich přidání. Obě tabulky obsahují cizí klíče pro přizování otázek k učitelům a odpovědi k otázkám viz. diagram.

3.5.3 Relace:

Poslední tři tabulky vytváří relace potřebné pro správu dat. Patří mezi ně `teacher_subject`, `enrollments` a `student_question`. Tabulka `teacher_subject`: Tato tabulka slouží k zachycení vztahu mezi vyučujícími a předměty. Slouží jako orientační bod v každé otázce. To znamená, že každý učitel má možnost pokládat otázky pouze na předměty které učí. Je to funkce nezbytná pro zachování pořádku ve zpětné vazbě. Tabulka `enrollments`: Tato tabulka slouží k evidenci, který student je zapsán na který předmět a od kterého učitele. Využívá k tomu cizí klíče `student_id` a `teacher_subject_id`. To umožňuje správu studentských zápisů, jinými slovy to slouží k filtrování otázek na které může daný student odpovídat. A to v závislosti na tom jestli je a nebo není konkrétní student zapsán na kurzu, ke kterému byla otázka položena. Tabulka `student_question`: Zaznamenává relace mezi studenty a otázkami, na které odpověděli. Je potřebná pro zajištění toho, aby mohl student odpovědět na každou otázku pouze jednou, přičemž by si zároveň zachoval anonymitu. Většina dat se tedy bude muset do databáze doplňovat přímo.



Obrázek 3.1: Databázový diagram

Závěr

Práce na tomto projektu pro mě byla něčím úplně novým. Přestože jsem se v tom ze začátku docela ztrácel a moc jsem netušil, co všechno budu muset udělat. Celkově si myslím, že se mi podařilo dosáhnout většiny cílů, které jsem měl. Aplikace umožňuje definovat otázky, sbírat odpovědi a analyzovat data z minulých let. Žáci mohou odpovídat pouze na otázky, které jsou jim určeny. Učitelé mají možnost otázky nejen přidávat ale i odebírat. To ale jen v případě, že na danou otázku ještě není přidána jediná odpověď. To mi totiž přišlo jako objektivnější řešení pro získávání zpětné vazby, aby si učitelé nemohli mazat otázky obsahující kritiku. Ted' zpětně ale přemýšlím, že by mělo vedení školy nějakou formu regulace obsahu mít. Jsem si samozřejmě vědom některých nedokonalostí, kterých jsem se ve svých řešení dopustil. Asi největším prohřeškem bylo zařízení anonymity pomocí databázové tabulky `student_question`, které anonymitu zajišťuje pouze při větším množství odpovědí na danou otázku. Další nedokonalostí je určitě struktura mého kódu, který jsem mohl napsat lépe. Samotná práce pro mě bylo velkou zkušeností a doufám, že své nově nabitě zkušenosti budu moci ještě někdy využít. Vlastně všechno, co jsem v projektu dělal až na psaní html kódů pro mě bylo novinkou. Od navrhování databáze až po implementaci jednotlivých funkcí v Python Flask. Také jsem se setkal s vývojem webových aplikací z hlediska autentizace uživatelů pomocí google oauth.

4. Citace

4.1

Mark Otto, J. T. (n.d.). Introduction. Bootstrap.

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

4.2

MySQL Workbench. MySQL. (n.d.).

<https://dev.mysql.com/doc/workbench/en/>

4.3

Welcome to flask. Welcome to Flask - Flask Documentation (3.0.x). (n.d.-a).

<https://flask.palletsprojects.com/en/3.0.x/>

4.4

GfG. (2023, December 1). Flask Tutorial. GeeksforGeeks.

<https://www.geeksforgeeks.org/flask-tutorial/>

Seznam obrázků

2.1	login.html	5
2.2	myactivequestions.html	6
2.3	addquestion.html	6
2.4	myanswers.html	7
3.1	Databázový diagram	11

Seznam tabulek