

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Sabína Ságová	
Název práce	Webová aplikácia na precvičovanie databázových dotyov	
Rok odevzdání	2024	
Studijní program	Informatika	
Specializace	Obecná informatika	
Autor posudku	RNDr. Jakub Klímeck, Ph.D.	Oponent
Pracoviště	KSI MFF UK	

K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání			X	
Rozsah práce <small>... textová i implementační část, zohlednění náročnosti</small>			X	
Zadání má běžnou obtížnost, avšak lze k němu přistoupit různě důsledně. Technicky splněno bylo, ovšem kvalita zpracování je ve srovnání s úrovní jiných bakalářských prací nedostatečná. Zcela chybí standardní softwarově-inženýrský přístup k vývoji software, samotný kód pak vykazuje řadu závažných prohřešků proti dobré praxi. Uživatelské rozhraní má i přes svoji jednoduchost k použitelnosti také stále daleko.				

Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <small>... jazyková úroveň, typografická úroveň, citace</small>		X		
Struktura textu <small>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</small>				X
Analýza				X
Vývojová dokumentace			X	
Uživatelská dokumentace		X		

Textová část práce obsahuje pouze popis již existující implementace. Obsahuje:

1. Přehled podobných aplikací,
2. Stručný popis existující implementace (že sestává ze 4 stránek),
3. Příručka studenta a učitele, tj. uživatelské dokumentace,
4. Přehledu základních použitých technologií a formátů - HTTP, REST, JSON apod.,
5. Popis adresářové struktury projektu, tj. stručnou vývojářskou dokumentaci,
6. Výsledky uživatelského testování.

Zcela chybí základní softwarově-inženýrský přístup k vývoji SW, kde by na začátku byla provedena analýza uživatelských rolí, požadavků, případů užití, následoval by návrh s architekturou řešení a mockupy, dále by teprve následovala část implementace s popisem zvolených technologií, jejich alternativ, a zdůvodněním jednotlivých voleb, a dále také informace o tom, jak je software testován.

Text zkrátka popisuje něco, co "nějak bylo naprogramováno", což je nedostatečné.

Drobnost: nadpis na straně 31 obsahuje překlep: technógií

Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu ... <i>architektura, struktury a algoritmy, použité technologie</i>				X
Kvalita zpracování ... <i>jmenné konvence, formátování, komentáře, testování</i>				X
Stabilita implementace		X		

Návrh řešení v práci zcela chybí, v textu práce je pouze popis již existující implementace. Kód pak vykazuje celou řadu nedostatků. Namátkou:

1. Správnost či špatnost SQL dotazu v úkolech se hodnotí na základě zkoumání stejnosti výsledků. To je z edukativních důvodů nedostačující. Například "Vyber všechny studenty" je vyhodnoceno správně jak v očekávané variantě `SELECT * FROM studenti`, tak v každé jiné, která vrací stejné výsledky jen náhodou, například `SELECT * FROM studenti WHERE id_student < 6`.
2. Dále v dnešní době je poněkud nepříjemné, když se dotaz píše do obyčejného textového pole, kde chybí dnes běžné věci jako zvýrazňování syntaxe, našeptávání, apod.
3. Zvláště pak působí napevno napsané ID uživatele "1" na řádku 84 v `ukoly.html`. ID 1 má zřejmě zvláštní funkci, vyskytuje se i v dotazu na řádku 102 v `index.py`, kde se úlohy uživatele 1 vrací nezávisle na konkrétním ID uživatele. Vysvětleno to ale nikde není. Navíc to znamená, že aplikace vrací vždy pouze úkoly uživatele 1, ostatní uživatelé mohou úkoly přidávat pomocí API, ale na stránce se nikdy nezobrazí, pokud je nevytvořím pod uživatelem 1.
4. Texty uživatelského rozhraní pak jsou přímo v kódu místo v extra souboru, kde by mohly být například snadno přeložitelné.
5. Backend není nijak zabezpečený, API může volat kdokoli odkudkoliv. Zadáním `https://slt.ufal.mff.cuni.cz/sabina-api/ukoly/id_uzivatele/1` do prohlížeče tedy vidím všechna zadání i řešení.
6. Zrovna tak HTTP POST pro přidání úkolu může udělat kdokoli, a znamená natvrdo zápis do databáze, bez ověřování existence použitého uživatelského ID (které ani existovat nemusí, stejně se úloha zapíše se zadaným ID).
7. Port, na kterém backend běží, je napevno napsán v kódu místo konfigurace, stejně jako třeba e-mail odesílatele e-mailů s potvrzením registrace.
8. Přihlašování uživatele (ř. 48 v `index.py`) je pak v podstatě jen na okrasu - v případě správného jména a hesla pouze vrátí ID uživatele. Pokud toto ID zjistím jinak, tak toto přihlašování k ničemu nepotřebuji a rozhodně neplní jakoukoliv bezpečnostní funkci.
9. Pole pro zadání SQL dotazů v UI jsou nesmyslně malá a nelze je ani rozšířit pomocí standardních funkcí prohlížeče.
10. Výpisy tabulek z databáze jsou pro změnu miniaturním písmem, přitom místa na obrazovce je dost.

Celkové hodnocení Neprospěl(a)
Práci navrhuji na zvláštní ocenění Ne

Datum 22.5.2024

Podpis