

## I. IDENTIFIKAČNÍ ÚDAJE

<b>Název práce:</b>	Tvorba webové aplikace pro projekt Viskalia
<b>Jméno autora:</b>	Bc. Adam Kulhavý
<b>Typ práce:</b>	<input type="text"/>
<b>Fakulta/ústav:</b>	<input type="text"/>
<b>Katedra/ústav:</b>	Katedra geomatiky
<b>Oponent práce:</b>	Jáchym Čepický
<b>Pracoviště oponenta práce:</b>	OpenGeoLabs s.r.o.

## II. HODNOCENÍ JEDNOTLIVÝCH KRITÉRIÍ

<b>Zadání</b>	<input type="text"/>
<i>Hodnocení náročnosti zadání závěrečné práce.</i>	
Zadaný obsah práce lze považovat za v praxi standardní. Zadání lze uchopit pomocí různých technologií a různých přístupů. Pro úspěšné dokončení práce je potřeba, aby student zvládl celou řadu technologií a stal se tolik vyhledávaným „full-stack“ vývojářem.	
<b>Splnění zadání</b>	<input type="text"/>
<i>Posudte, zda předložená závěrečná práce splňuje zadání. V komentáři případně uveďte body zadání, které nebyly zcela splněny, nebo zda je práce oproti zadání rozšířena. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.</i>	
Zadání lze v principu považovat za splněné na úrovni MVP (Minimum viable product). Pro nasazení v praxi by byla potřeba ještě celá řada změn. Takto ale lze na práci stavět další práci. Práci se mi nepodařilo zprovoznit ani po odstranění jednoho problému (startup.sh nebyl součástí docker containeru a nebyl spustitelný). Těžko říct.	
<b>Zvolený postup řešení</b>	<input type="text"/>
<i>Posudte, zda student zvolil správný postup nebo metody řešení.</i>	
Způsob řešení bych považoval za v celku standardní. Obecně se nedoporučuje pouštět do administrativního rozhraní frameworku Django běžné uživatele (byť po přihlášení). Potenciální problémy se ale v práci řeší zavedením uživatelských skupin.	
<b>Odborná úroveň</b>	<input type="text"/>
<i>Posudte úroveň odbornosti závěrečné práce, využití znalostí získaných studiem a z odborné literatury, využití podkladů a dat získaných z praxe.</i>	
Nároky na fullstack development jsou vysoké: je potřeba zvládnout technologie serverové strany (Python, Django), klientské strany (HTML, JavaScript), návrh a správu databáze, tvorbu datového modelu. Když už se zdá, že je všechno naprogramované, nastává fáze nasazení (deployment), což v sobě obsahuje technologie Docker a docker-compose. Správu kódu v prostředí verzovacího systému (Git) je také potřeba započítat. Z tohoto pohledu lze konstatovat, že práce splňuje standardní požadavky na zvládnutí těchto technologií na úrovni, kdy je student schopen tyto znalosti dále samostatně rozvíjet.	
<b>Formální a jazyková úroveň, rozsah práce</b>	<input type="text"/>
<i>Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku.</i>	
Nemám výhrady ani k formální, ani k jazykové úrovni práce.	
<b>Výběr zdrojů, korektnost citací</b>	<input type="text"/>
<i>Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení závěrečné práce. Charakterizujte výběr pramenů. Posudte, zda student využil všechny relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků a úvah, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami.</i>	

Práce se zabývá vývojem webových aplikací a z mého pohledu je relevantní citovat co nejvíce online zdrojů. Přijde mi ale zvláštní, že v práci zabývající se frameworkem Django není ani jednou použita přímá citace tohoto frameworku i s přihlédnutím k tomu, že správná citace je popsána přímo v dokumentaci <https://docs.djangoproject.com/en/3.2/faq/general/#how-do-i-cite-django>

To samé platí o diskutovaném frameworku Flask, na který je odkázáno pouze nepřímě.

Po mém soudu je rešeršní část odcitována dostatečně (s výše zmíněnými výhradami), praktická část je bez výhrad a vychází především z oficiální dokumentace.

### **Další komentáře a hodnocení**

*Vyjádřete se k úrovni dosažených hlavních výsledků závěrečné práce, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, publikačním výstupům, experimentální zručnosti apod.*

\* V části věnované frameworku Flask se v práci úplně pomíjí tvorba ORM pomocí nějaké knihovny třetí strany (např. SQLAlchemy) a je zdůrazněna pouze možnost „ručního“ mapování dat na datový model v databázi

\* Zarazila mě věta (str 22): „Data do databáze šla přidávat, zobrazovat je a mazat, ale při editaci se po uložení nepřepsala aktuální data v databázi.“ - to je u Django frameworku neobvyklé. Můžete to prosím blíže rozvést? Souvisí to s nastavením práv pro zápis do souborů/databáze?

\* Pouze upozornění: soubory v migrations/ jsou považovány běžně za součást projektu. Je správné mít je uložené do verzovacího systému. To umožní sdílet práci s ostatními kolegy v týmu (v budoucnosti). Současné řešení neumožňuje jednoduše dělat změny v datové struktuře projektu.

\* Vlastnímu kódu chybí použití nějakého „linter“ nástroje: Dlouhé řádky, nedostatečně oddělené bloky, dlouhé funkce. Je správné používat nástroj (ať už vestavěný v editoru nebo externí), který hlídá a staticky kontroluje vzhled zdrojového kódu.

\* Dokumentace (README.md) existuje – to hodnotím velmi kladně. Jedna chyba: skript `startup.sh` nebyl nastaven (v příloženém ZIP archivu) jako spustitelný a nebyl součástí containeru. Po přidání do containeru se aplikace sice spustila ale ve výpisu se objevila chyba

```
viskalia_editapp_1 | standard_init_linux.go:228: exec user process caused: no such file or directory
```

Závěr: bohužel se mi z příloženého ZIP archivu nepodařilo bez většího úsilí aplikaci zprovoznit. Databáze a administrační nástroj pro databázi fungují.

\* V práci je napsáno, že zdrojový kód je udržován na serveru Gitlab, odkaz jsem ale nenašel?

### **III. CELKOVÉ HODNOCENÍ, OTÁZKY K OBHAJOBĚ, NÁVRH KLASIFIKACE**

*Shrňte aspekty závěrečné práce, které nejvíce ovlivnily Vaše celkové hodnocení. Uveďte případné otázky, které by měl student zodpovědět při obhajobě závěrečné práce před komisí.*

Předloženou závěrečnou práci hodnotím klasifikačním stupněm

**Způsob inicializace modelů (inspectdb + šablona models\_django.py) mi nepřijde vhodný. Nebylo by přeci jenom lepší, použít inspectdb jen napoprvé a následně už upravovat modely ručně a pomocí nástrojů makemigrations && migrate ?**



## POSUDEK OPONENTA ZÁVĚREČNÉ PRÁCE

Datum: 15.6.2021

Podpis:

A handwritten signature in blue ink, reading 'Jáchym Čepič', is positioned to the right of the 'Podpis:' label. The signature is written in a cursive style.