

ZÁVĚREČNÁ MATURITNÍ PRÁCE

GYMNÁZIUM JANA KEPLERA
Informatika

Maturitní závěrečný projekt

APstatus: Aplikace k monitorování IOT zařízení na GJK

GYMNÁZIUM JANA KEPLERA
Informatika

Prohlášení

Tímto prohlašuji, že jsem práci vypracoval sám a veškeré použité zdroje jsou uvedeny v seznamu literatury. Tímto prohlášením potvrzuji, že jsem neuvedl texty z převzatých zdrojů, aniž bych je citoval.

Úvod do problematiky

Zaměřením práce bylo vytvořit, webovou aplikaci sloužící k monitoringu síťových prvků na Gymnáziu Jana Keplera, jako jsou například Routery, Switche, Acces-pointy, tiskárny a IP telefony.

Cílem bylo vytvořit službu, která bude přístupná přes webové rozhraní, bude nabízet možnost modifikace sady síťových prvků buďto přes konfigurační software nebo přes webové rozhraní chráněné přihlášením. A zároveň umožnit jednoduché nasazení aplikace pomocí technologie Docker nebo obdobné služby.

Práce je přínosná k chodu našeho gymnázia, ale také pomohla mě, autorovi, ve zdokonalení se programování a znalosti v oboru síťové infrastruktury.

Text Práce

Analýza

Projekt mi byl přidělen na základě požadavků rozsahu závěrečného maturitního projektu a potřebu pro plně funkční aplikaci k monitorování IOT prvků na GJK.

Po konzultaci s vedoucím mého projektu Františkem Šimordou, jsme si upřesnili, že je potřeba vytvořit aplikaci zobrazující vizualizaci dostupnosti IOT prvků v řádu posledních dvanácti hodin ve webovém rozhraní. Zároveň mi byli Františkem Šimordou doporučeny platformy Django a Docker pro vývoj backendu aplikace a její následnou implementaci.

Co se analýzy týče, začal jsem návrhem databáze potřebnou pro správné skladování potřebných informací a jednoduchou editaci. Z několika variant jsem zvolil verzi, kde skladuji všechny informace v jedné tabulce s čtrnácti hodnotami. Rozhodl jsem se také pro použití převážně jazyku Python (ve verzi 3.7.2), databázového systému sqlite3 a Django (2.1.1) jako prostředku k vytvoření a provozování potřebného backendu a databáze.

Implementační zajímavosti

Na začátku vývoje, sudující zároveň potřebnou dokumentaci, jsem nejprve experimentoval s platformou Docker, která pro mne v danou chvíli byla velkou neznámou. Největším problémem na počátku vývoje pro mne byla implementace Django skrze Docker, kde jsem zápasil s nastavením firewallu, který mi zabraňoval Django přes Docker správně implementovat. Rozhodl jsem se tedy pro postup rychle navrhnout správu databáze přímo v Pythonu. Naprogramoval jsem tedy pár jednoduchých funkcí k vytvoření a editaci potřebné databáze. Naleznete je v mém projektu ve verzovacím systému Github ve verzi 1.1 a starší, pro běh finální aplikace nebyli potřeba a byli proto odebráni ve snaze udržet kód co nejlehčí.

Nyní již ale vím, že kdybych býval byl Django implementoval lokálně mimo Docker hned na počátku a později Dockerizoval, psaní těchto funkcí bych se zcela vyhnul, jelikož Django nabízí daleko intuitivnější způsob managementu databází již v základu.

Průběh testování a vývoje

Po zhotovení lokální funkční verze aplikace APstatus a následné zdařilé implementaci Django přes Docker o několik desítek commitů později jsem začal pracovat na propojení webové stránky a backendu vytvořené Djangem, s databází a jádrem samotné aplikace (main.py). Ta podle mého názoru proběhla relativně zdařile a bez větších komplikací a mohl jsem se tak vrhnout na tvorbu samotného zobrazení vizualizace dostupnosti IOT zařízení na webové stránce. Tu jsem zhotovil pomocí chytré implementovaných Django templatového systému a HTML/Css. Výsledkem byla tabulka barevně graficky znázorňující dostupnost daného síťového prvku za posledních dvanáct hodin.

Poslední den vývoje projektu jsem strávil testování, objevil jsem několik chyb (bugů), například u zobrazování některých krajních hodnot, výpočtu času ve vizualizaci hodnot a parsování výstupního textu z jádra aplikace. Po konzultaci s panem Šimordou jsem dále objevil nekompatibilitu mnou zvoleného fontu s některými zařízeními, potřebu pro přidání další hodnoty do databáze síťových prvků, uživatelem nastavitelné jméno dotyčného síťového prvku, pro zjednodušení čtení vizualizace.

Následně jsme spolu finálně odzkoušeli implementaci kompletní aplikace na nezávislý systém, která proběhla bez komplikací. Po napravení všech chyb a požadavků ve fázi testování byla aplikace APstatus hotova.

Závěr

Rád bych poděkoval vedoucímu mé práce Františku Šimordovi a mému vyučujícímu Karlovi Jílkovi za flexibilní, vřelý a ochotný přístup, kterým mne podporovali v úspěšné tvorbě.

Myslím, že ve výsledku jsem nenarazil na žádný nepřekonatelný problém, a vše dobře dopadlo. Navíc jsem si značně zdokonalil znalost Django, osvěžil zkušenosti s Pythonem a naučil se používat Docker. K tomu jsem navíc vyvinul funkční aplikaci s reálným využitím, což vede k mému neskonalému potěšení z podle mne dobře odvedené práce.

Seznam použitých materiálů a literatury

- Webové stránky a použitá literatura
 - stackoverflow.com
 - w3schools.com
 - runnable.com
 - freecodecamp.org
 - sqlitetutorial.net
 - python.org
 - youtube.com/user/thenewboston
 - thenewboston.com
 - github.com
- Python:
 - python.org
- Django:
 - djangoproject.com
- Docker:
 - docker.com

Přílohy

- Elektronické forma projektu na [GitHub.com](https://github.com)