Západočeská univerzita
Fakulta aplikovaných věd
Katerdra Kybernetiky

../pic/fav.jpg

Dokumentace distribuci projeku ITE-YELLOW

Martin Hamar, Radek Kaupe, Samuel Kokoška

KKY/BSOI Datum: 10. ledna 2025

1 Úvod

V rámci předmětu KKY/BSOI jsme si vytvořili a nastavili virtuální stroj s operačním systémem Linux. Tento stroj je momentálně používán k provozu applikace vyvíjené v rámci předmětu KKY/ITE. Zde je na aplikaci a zde je odkaz na Githubový repozitář, kde jsou k dipozici veškeré zdojové kódy aplikace jak části ITE, tak části BSOI. Tento referát by měl popsat virtuální stroj, jeho nastavení, činnosti a realizaci způsobů přihlášení do již zmíněné aplikace.

2 Řešení

2.1 Zálohování

Zálohování databáze bylo implementováno pomocí nástroje Borgbackup, specificky funkcí prune pro automatizaci celého procesu k ukládání každodenní zálohy na účty skokoska a radek, v případě poruchy či potřeby vrátit databázi do předchozího stavu tedy existují dvě zálohy.

Na každém účtě je složka "backup"obsahující zašifrovanou zálohu pod defaultním heslem "gobackImadeamistake".

2.2 Provoz distribuované aplikace

Součásí aplikace je PostgreSQL databáze a dva python skripty, jeden z nich řeší MQTT subscribera, abychom mohli data příjímat, validovat a ukládat a druhý tvoří backend naší webové aplikace. K založení PostgreSQL datbáze, byl také vytovřen python skript, který stačí spustit pouze jedno, aby vytvořil tabulky. Skripty na subscribera a backend musí běžet neustále, aby aplikace mohla být v provozu. Tyto skript běží současně pomocí tmuxu. Tmux je 'Terminal Multiplexer', který umožňuje rozdělit obrazovku terminálu na více panelů a vytváření 'sezení' (sessions), které mohou běžet i na pozadí, poté co se člověk odpojí. Subscriber a backend spolu 'komunikují' pouze přes databázi, ke které mají oba skripty přístup.

ZAJISTI AUTOMATICKY SPUSTENI PO RESTARTu

Momentálně logování výstupu je k vidění pouze v rámci tmux session, kde jsou nastavené výpisy za běhu obou skriptů, aby mohlo dojít ke kontrole kdykoliv některým členů z týmu. Nasazení verzí se může provést pomocí programu WinSCP, kterým se nové soubory nahrají na virtuální stroj.

3 Přihlášení a biometrická autentizace

Při implementaci jsem využil učiteli dodaných zdrojů, ve formě skriptů `faceid_server.py`, `extract_embeddings.py`a `train.sh`.

Rozšíření backendu o FaceID
V prvním skriptu se nacházejí Tornado Handleři, jeden zajišťující komunikaci s frontendem v rámci posílání a ukládání fotek a druhý v rámci detekci obličeje ve fotce. Předtím oba dva handleři ???fungovali??? v rámci jedné URL adresy. Já je rozdělil. Aplikace funguje následovně:

1. Nepřihlášený uživatel se může buď přihlásit přes uživatelské jméno a heslo (vstupní stránka), přihlásit se přes FaceID nebo se zaregistrovat
2. Po registraci musí uživatel čekat na schválení adminem, jinak se nepřhlásí (ani v případě, že již fotky má uložené v rámci biometické autentizace)
3. Přihlášený uživatel má přístup k celé aplikaci, včetně možnosti nafotit si své fotky pro FaceID a natrénovat model pro autentizaci obličejů

Uživatelské účty a identity
Máme celkem 9 uživatelských účtů pro příhlášení, pro:

- 4 vyučující
- 3 členy týmu
- 1 admin účet, ke kterému mají přístup všichni členové týmu

Identit je 8. Jsou vytvořené pro všechny členy týmu, jeden kategorie 'unknown' (aby se 'nové' tváře nemohli přihlásit, jiank by je systém identifikoval jako validního uživatele) a 3 identity učitelé, aby mohlo dojít k právě zmíněné identity 'unknown'. Kategorie 'unknown' má celkem 17 fotek, z toho 3 neobsahují lidský obličej, aby se neuronová síť naučila přiřazovat neplatné fotografie neautorizovanému uživateli. Každý z členů má 10 až 25 fotek, které si každý sám nafotil. Od každého vyučujícího jsme dostali 10 - 25 fotek. Každý s různým počtem, aby šlo testovat úspěšnost identifikace. Rozhodovací práh je nastaven na alespoň 90% pravděpodobnost identifikace uživatele. Je nutné podotknout, že identita 'unkown' není vůbec uložená v databázi a nemá možnost se do aplikace přihlásit.

Funkčnost přihlašování
V případě, že se člověk přihlásí, je mu nastaven Json Web Token (JWT). Ten řeší zda je uživatelská session je validní či nikoliv. Pokud jsou uživatelská credentials v pořádku a uživatel je zaregistován a schválený adminem je mu JWT uložen jako secure cookie. Všechny další žádosti nyní automaticky kontrolují zda je JWT neexpiroval nebo s ním nebylo manipulováno. JWT má totiž nastavenou donbu expirace jedné hodiny a obsahuje 'podepsán' tajným klíčem, který je znám pouze serveru.

Odhlašování
Pro odhlašování byl vytvořen Tornado Handler, který vymaže token obsahující cookie a přesměruje uživale zpět na přihlašovací stránku.

4 Závěr