

POSUDEK BAKALÁŘSKÉ PRÁCE

Autor: Jiří Jindra
Název: Aplikace Půjčovna aut – Uživatelská strana
Posudek vypracoval: oponent práce RNDr. Ondřej Žára

Bakalářská práce se věnuje rozboru a implementaci klientské strany webové aplikace **Půjčovna aut** pro správu a rezervaci sdílených vozidel. Je součástí většího projektu, jehož serverovou protistranu implementoval Artur Sarhsjan. Výsledná aplikace je v rámci možností funkční a se serverem komunikuje pomocí HTTP REST API. Je značná škoda, že součástí práce není přehled metod tohoto API.

Text práce je na dobré jazykové úrovni, úprava dokumentu je v pořádku. K práci s literaturou a citacemi nemám výhrad. Je škoda, že nikde v sedmdesáti stránkách není ani jedna ukázka UI (screenshot) výsledných stránek. Kapitola o testování aplikace detailně rozebírá proběhlé kvalitativní uživatelské testy, ale ani teoreticky nezmiňuje možnost strojových/automatizovaných testů.

K praktické části práce mám celou řadu připomínek, které lze shrnout do těchto hlavních bodů:

1. **Chaotický návrh a struktura aplikace.** Autor práce využil nezvyklý přístup na půl cesty mezi tradiční sadou stránek a *single-page aplikací (SPA)*. Bohužel tak vznikl produkt, který si z obou těchto přístupů bere to horší: při uživatelské navigaci dochází k pomalé top-level navigaci (koncept běžných stránek), nicméně získaný dokument neobsahuje relevantní data, neboť se jedná o inicializační kód pro redundantně stahovanou SPA, která svá data získá až se zpožděním pomocí JavaScriptových dotazů na REST API backendu. Taková forma řešení není vhodná a ani v praxi ji nikde nevidáme.

Samotný klientský kód je přitom ohromný (přes 200 KB JavaScriptu, tranzitivní závislosti třetích stran v přibaleném archivu `node_modules.zip` mají neuvěřitelných 500 MB), takže by dávalo smysl minimalizovat počet plnohodnotných načtení aplikace. Zmiňovaný archiv je výstřednost sama o sobě; taková data do repozitáře v žádném případě nepatří a dle slov autora je jejich přítomnost „*kvůli problémům*“.

2. **Chybějící či slabé ospravedlnění klíčových technologických rozhodnutí.** Zdrojový kód aplikace působí jako skládanka *aktuálně moderních buzzwordů*, se kterými se autorovi možná pohodlně pracovalo, ale jejich vliv na kvalitu produktu považuji za velmi diskutabilní. Text nevysvětluje, proč bylo nutné použít SPA (viz předchozí odstavec); stejně tak není popsán důvod použití knihovny *Axios* v kontrastu s moderní vestavěnou funkcionalitou *fetch*. Součástí produktu je práce s HTTP cookies (opět pomocí nadbytečné knihovny třetí strany), ovšem tato funkcionalita je nesmyslná. Bezpečnostní model aplikace totiž stojí na konceptu serverových sessions, které jsou spravovány mimo logiku klientského kódu. Pro potřeby SPA je tedy bezpředmětné ukládat data do cookies. Součástí zdrojových kódů je soubor `serviceWorker.js`, který patrně není dílem autora a přitom není zmíněn nikde v textu práce.
3. **Menší a nesouvisející výhrady ke zdrojovému kódu jako takovému.** Uživatelské heslo je při přihlášení posíláno v URL metodou GET. Absolutní adresa URL aplikace je napevno zmíněna na mnoha místech, takže ji není možné přímočaře provozovat na jiném portu či IP adrese. Na téměř pěti tisících řádek JavaScriptu není ani jeden výskyt dokumentačního komentáře.

S ohledem na popsané nedostatky navrhuji práci ohodnotit známkou **C – dobře**.