

Technologie pro publikování na webu 2

Semestrální projekt – webová aplikace

Identifikační údaje

Jméno a příjmení:
Školní login:
Studijní obor:
Akademický rok:
Cvičení:

| |
|------------------------|
| David Továrek |
| tovarda1 |
| Aplikovaná Informatika |
| 2021/22 |
| St 08:15-09:50 |

Souhrnné informace

Název projektu:
Anotace projektu:

| |
|--|
| The Relay |
| The Relay je webová aplikace dovolující registrovaným uživatelům nakupovat a prodávat zboží. Aplikace je vytvořena pomocí front-endové knihovny React. |

A. Uživatelská dokumentace

Popište používání aplikace.

(Spuštění: po spuštění databáze je nutné spustit backend server pomocí „node server.js“ ve složce the-relay-server. Front-end aplikace je pak spuštěn příkazem „npm start“ ve složce the-relay-client)

Nepřihlášený uživatel si může přečíst uvítací text, popis stránky a licenční ujednání. Pro samotné vytváření a prohlížení inzerátů je nutné se přihlásit, nebo vytvořit účet přes položku „login“.

Po vytvoření účtu se uživatel může přihlásit a prohlížet inzeráty pod položkami „buy“ a „sell“. V případě zájmu může uživatel kliknout na tlačítko „I am interested!“, které otevře e-mailový klient s předpřipravenou zprávou pro inzerenta. Inzeráty lze filtrovat pomocí ovládacích prvků v horní části stránky a to podle kategorie, ceny, nebo textu (vyhledávání v názvu a popisu). Filtr lze odstranit tlačítkem reset.

Plovoucí tlačítko v pravé dolní části obrazovky slouží k vytvoření vlastního inzerátu (kategorie prodej/nákup se řídí aktuální stránkou). Cena v případě prodeje znamená cenu produktu, v případě nákupu pak maximální cenu, kterou je uživatel ochoten zaplatit za poptávané zboží.

Pro prohlížení, úpravu a mazání vlastních inzerátů slouží položka „My listings“.

B. Dokumentace architektury

Popište klíčové koncepty a principy, na základě kterých byla navržena architektura aplikace. Popište logiku fungování aplikace.

Aplikace byla navržena na principu tzv. MERN stacku. Skládá se z front-endu využívajícího knihovnu React, který je rozdělen na komponenty. Ty jsou vzájemně skládány dohromady pro dynamickou tvorbu stránky. Pro navigaci se využívá React router, který na základě url stránky zobrazuje definované komponenty (cesty v App.js).

Jako back-end slouží server vytvořený pomocí knihovny Express, který interaguje s MongoDB databází pomocí knihovny Mongoose. Ve složce „routes“ jsou vytvořeny samostatné routery pro manipulaci s příslušným zdrojem (účty, inzeráty, kategorie). Samostatně pak existuje autorizační middleware, který je vyžadován pro obsluhu některých koncových bodů.

C. Dokumentace API

Popište koncové body aplikace a jejich obsluhu.

Koncové body pro uživatelské účty:

/account/add - POST

Slouží pro vytvoření účtu

Očekává objekt s vlastnostmi username, email a password

Vrací kód 201 a objekt účtu při vytvoření – 400 pro neplatný dotaz (duplicitní e-mail), 500 pro chybu serveru

/account/login - POST

Slouží pro přihlášení

Očekává uživatelské jméno a heslo

Vrací 200 a JWT token při úspěšném přihlášení – chyby 403 pro špatné heslo, 404 pro špatný účet

/account/logout - GET

Slouží pro odhlášení

Očekává autentizaci uživatele k odhlášení

Vrací 200 při odhlášení, 404 pokud nebyl nalezen uživatel s daným id

/account/me - GET

Slouží pro ověřování platnosti přihlášení

Vyžaduje autentizaci

Vrací uživatelské jméno

Koncové body pro inzeráty:

/listing/getFiltered - POST

Slouží pro získání inzerátů dle filtru

Vyžaduje objekt MongoDB filtru

Vrací 200 a seznam inzerátů – 404 v případě nenalezení

/listing/getUserListings - GET

Slouží pro získání inzerátů uživatele

Vyžaduje JWT autentizaci

Vrací 200 a seznam inzerátů – 404 v případě nenalezení

/listing/update - POST

Slouží pro úpravu inzerátu

Vyžaduje autentizaci, id a objekt inzerátu

Vrací 201 při úspěšné úpravě – 500 pro chyby

/listing/delete - POST

Slouží pro mazání inzerátu

Vyžaduje autentizaci a id mazaného inzerátu

Vrací 200 a „Deleted“ při smazání – 500 pro chyby

/listing/add - POST

Slouží pro vytvoření inzerátu

Vyžaduje autentizaci a objekt inzerátu

Vrací 201 při úspěšném vytvoření – 500 pro chyby

Koncové body pro kategorie:

/category/getAll - GET

Slouží pro získání seznamu kategorií

Vrací 200 a seznam kategorií – 500 pro chyby

D. Dokumentace databáze

Popište model jednotlivých zdrojů a vztahy mezi nimi.

Databáze obsahuje tři kolekce: účty, inzeráty a kategorie.

Kategorie obsahují pouze jméno a id, slouží pro filtrování inzerátů, jedná se pouze o String hodnoty. Uživatelský účet se skládá z jména, hesla, e-mailu a seznamu tokenů. Všechny tyto položky jsou povinné, e-mail je validován knihovnou validator. Model má metodu pro generování tokenů a má vztah s Inzerátem.

Inzerát je objekt s odkazem na uživatele, který ho vytvořil, názvem, popisem, kategorií, cenou a bool hodnotou určující, zda se jedná o prodej, nebo nákup. Všechny položky jsou povinné, cena je validována a nesmí být záporná (může však být 0 pro případy věnování zboží).

E. Kontrolní checklist

Tento protokol semestrálního projektu má obsahovat:

- ☐ Uživatelská dokumentace
- ☐ Dokumentace architektury
- ☐ Dokumentace API
- ☐ Dokumentace databáze