WWW-sovellukset
Sakari Laine
0418598
Harjoitustyön dokumentaatio

Dokumentaatio harjoitustyöstä (Movie and TV show subscriber)

Alustus

Valitsin kyseisen aiheen, koska halusin luoda itselleni jotain hyödyllistä. Ennen odotin elokuvista DVD-versioita ja sarjoista uusia jaksoja, mutta aina ei pysty muistamaan ulkoa, milloin mikäkin DVD ja jakso on tullut ulos, jota auttamaan tämä projekti on tehty. Koko projektin tein yksinäni.

Palvelun ominaisuuksia

Palvelusta löytyvät seuraavat ominaisuudet:

- Rekisteröityminen, kirjautuminen, Steam-kirjautuminen (OpenID:n avulla)
- IMDb API:n käyttäminen hakutuloksissa
- OMDb API:n käyttäminen TV-sarja- ja elokuvatietojen hakemisessa perustuen IMDb:n palauttamiin tuloksiin. OMDb API:sta haetaan myös TV-sarjojen kausien jaksot.
- Mahdollisuus merkitä elokuvia/tv-sarjoja seurantalistalle
- Mahdollisuus merkitä tv-sarjojen jaksoja ja kausia katsotuiksi

- Oman seurantalistan hakeminen
- Seurantalistan seulominen siten, että näyttävät vain elokuvat, joissa on DVD-versio tullut, ja tv-sarjat, joissa on jaksoja joita ei ole vielä katsonut
- Poster-kuvien tallentaminen palveluun varmistaakseen kuvien saatavuuden (IMDb ei hyväksy hotlinkkausta)
- Sarjojen viimeisimmän jakson tietojen (episode, season) tallentaminen json-tiedostoon ja sieltä lukeminen

Palvelun täyttämät vaatimukset

Palvelusta löytyvät seuraavat vaatimukset harjoitustyön vaatimuksista:

- Responsiivinen ulkoasu (skaalautuu pienemmille näytöille myös)
- Tietokannan käyttö tietojen tallennukseen (käyttäjä-taulu, followlist-taulu, seenlist-taulu)
- Käyttäjän autentikointi hasheineen ja suolineen (rekisteröityminen, kirjautuminen)
- Front-controllerin käyttö (ulkoisesti kaikki toiminnot tapahtuvat etusivulla)
- MVC-mallin mukainen sivusto (noudattaa MVC-mallia)
- Välimuistin käyttö (välimuistiin tallennetaan niin title-tietoja, tuotantokausien tietoja, viimeisimpiä jaksoja, käyttäjän seuraamia titlejä, käyttäjän katsottuja titlejä, jne.)
- SEO (hakukoneoptimoitu; w3c-validatorin mukainen sivusto, meta content, title info, meta keywords)
- Kolmannen osapuolen palvelun käyttö (Steam login)
- Toinen kolmannen osapuolen palvelun käyttö (IMDb API ja OMDb API)
- JSONin käyttö tiedon tallennuksessa (tallentaa tv-sarjojen viimeisimmät julkaistut jaksot lastepisodes.json-tiedostoon)
- jQueryn käyttö

- Kaikki grafiikka on tehty dynaamisesti CSS:llä ja muilla tekniikoilla (grafiikka on tehty sekä CSS:llä että jqueryllä)
- Ajax-ohjelmoinnin hyödyntäminen (lukee .js-tiedosto lukee .php-tiedostoista tietoa ja syöttää niihin)

Bonus:

- Kattava dokumentaatio

Yhteensä 51 pistettä.

Palvelun asentaminen

1. Ensiksi pura tiedostot haluamaasi kansioon.
2. (Vaihtoehtoinen) luo uusi tietokanta tai valitse haluamasi tietokanta
3. Asenna seuraavat PHP-pluginit:
    a. php5-mysql
    b. php5-curl
    c. php5-memcached
    d. php5
4. Muuta /inc/settings.php-tiedostoa täyttämään seuraavat tiedot:
    a. tietokannan nimi, käyttäjätunnus, salasana
    b. Steam API Key
        i. Mene https://steamcommunity.com/dev/apikey ja hanki sivustollesi oma, uniikki API-avain, ja liitä se settings.php
    c. Haluamasi suolaus “hash”-kohtaan (mahdollisimman sattumanvarainen)
    d. Domain-kohtaan oma domainisi
    e. Title1- ja Title2-tietoihin tekstit, jotka haluat näyttää sivun bannerissa
5. Avaa selaimessa “install.php”-tiedosto.
    a. Jos taulujen luominen onnistuu, niin **poista install.php-tiedosto** palvelimelta.

b. Jos ei ilmesty tekstiä tai tulee virheilmoitus, niin lisää manuaalisesti tietokantaan seuraavat taulut:
    i. HT_accounts
        1. id, int, primary key, auto increment
        2. username, text
        3. password, text
        4. steamid, text
    ii. HT_followList
        1. id, int, primary key, auto increment
        2. userid, int
        3. showid, text
    iii. HT_seenList
        1. id, int, primary key, auto increment
        2. userid, int
        3. showid, text
        4. season, int
        5. episode, int
    iv. Kun olet luonut edellä mainitut taulut, **poista install.php-tiedosto**.
6. Muuta seuraavat käyttöoikeudet “777”:ksi
    a. /img/posters/-kansio
    b. /data/-kansio
    c. /data/lastepisodes.json-tiedosto

Käytin itse apache-web-serveriä devausympäristönä, mutta koodin pitäisi toimia myös muilla alustoilla.

Palvelun tarkempi katsaus

Selitettynä palvelun toimintoja:

1. Käyttäjä yrittää kirjautua sisään painamalla “Login”-nappia.
    a. script.js-tiedoston #loginButton-nappiin liitetty kuuntelija lähettää username- ja password-kenttien sisällöt login-funktiolle
    b. login-funktio käyttää Ajax-ohjelmointia hyödykseen lähettämällä tiedot registration.php-tiedostoon
    c. login-funktio jää kuuntelemaan, mitä registration.php palauttaa
    d. registration.php-tiedostossa ensiksi validoitaan pyyntö ja sitten tarkastetaan syötteet ja muutetaan salasana
    e. registration.php tarkastaa tietokannasta, löytyykö tulosta
        i. jos löytyy, palautetaan “error”=false ja lisätään käyttäjän session-muuttujaan käyttäjänimi ja id (tietokannasta)
    f. mikäli jokin kohdista d tai e epäonnistui, palautetaan “error”=true ja virheviesti
    g. login-funktio saa viestin takaisin
        i. mikäli ei virhettä, päivitetään sivu, jolloin käyttäjä näkee itsensä kirjautuneena sisään
        ii. mikäli virhe, niin ilmoitetaan siitä

2. Käyttäjä hakee elokuvaa ja painaa “Search!”-nappia
    a. script.js-tiedostossa #searchButton-nappiin liitetty kuuntelija lähettää searchField-kentän tekstin searchFunction-funktiolle ja syöttää “default”-tekstin mukaan, jotta searchFunction tietää mitä toimintoa käyttää
    b. searchFunction hakee search.php-tiedostosta dataa syöttämällä saadun syötteen mukaan
    c. search.php hakee IMDb API:sta syötteen ehdotukset
    d. jokaista palautettua tietuetta kohtaan ensiksi tarkastetaan löytyykö Memcachedista ja jos ei, niin haetaan OMDb API:sta tarkemmat tiedot (käyttämällä IMDb ID:tä)

        i. mikäli löytyi OMDb API:sta haetullla IMDb ID:llä, tarkistetaan, että haetun tietueen tyyppi on tv-sarja tai elokuva (eikä esimerkiksi videopeli)
        ii. tallennetaan saadut tiedot Memcachediin 18 tunniksi
    e. luetaan Memcachedista kyseisen imdbid:n tiedot (jatketaan askeleita d-e kunnes korkeintaan 5 tulosta on löytynyt)
    f. palautetaan kaksiuloitteinen lista searchFunction-funktiolle
    g. searchFunction syöttää saadut tiedot titleConstructor-funktiolle
    h. titleConsctructor tekee uuden titleList-elementin (poistaen vanhan samalla) ja tarkastaa, että on löytynyt ja jokaista alkiota kohtaan hakee tiedon getfollow.php-tiedostosta siitä, seuraako käyttäjä kyseistä elokuvaa / tv-sarjaa
        i. getfollow.php tarkistaa pyynnön, tarkistaa onko tallennettu Memcachediin, jos ei, niin haetaan followList-taulusta tieto, seuraako käyttäjä tietuetta vai ei
        ii. palautetaan 1 tai 0 sen perusteella, löytyikö tulosta vai ei
    i. lisää #title-nimisen elementin, joka sisältää elokuvan / tv-sarjan tiedot, titleList-elementtiin ja titleList lisätään mainBodyyn

3. Käyttäjä painaa TV-sarja-elementtiä (#titleSeries)
    a. script.js tarkistaa, onko kyseinen tv-sarja valmiiksi avattu vai suljettu
        i. jos avattu, suljetaan se
    b. syötetään imdbid ja kyseinen titleSeries-objekti seasonList-funktiolle
    c. seasonList-funktio hakee getseen.php-tiedostosta viimeisimmän jakson kyseisestä sarjasta, jonka käyttäjä on nähnyt
        i. getseen.php hakee Memcachedista tiedon, onko käyttäjälle tallennettu kyseisen tv-sarjan kohdalla tietoa viimeisimmästä nähdystä jaksosta
        ii. jos ei, niin haetaan seenList-taulusta

iii. palautetaan joko löydetty tieto tai 0,0

d. seasonList-funktio hakee getseries.php-tiedostosta tuotantokausien tiedot
    i. getseries.php tarkistaa Memcachedista, onko tallennettuja tietoja
    ii. jos ei, niin haetaan OMDb API:sta tuotantokausien tietoja kunnes tietoja ei löydy
    iii. tallettaa jaksojen tiedot ja tarkistaa samalla onko päivämäärä mennyt vai ei
    iv. päivittää tarvittaessa lastepisodes.json-tiedoston
    v. tallentaa memcachediin sekä viimeisimmän jakson että koko tuotantokauden tiedot, molemmat 18 tunniksi
    vi. palauttaa tuotantokausista koostuvan listan seasonList-funktiolle
e. seasonList tekee uuden seasonList-elementin ja poistaa vanhan.
f. seasonList tarkistaa, onko tuotantokausi kesken, julkaistu vai julkaisematta
g. tekee uuden .season-luokkaisen elementin, joka sisältää tuotantokauden tiedot
h. lisää sen seasonListiin ja seasonListin.mainBody-elementtiin

4. Käyttäjä painaa “Follow”-nappia
    a. Mikäli käyttäjä ei ole kirjautunut sisään, script.js ohjaa käyttäjän goToLoginForm-funktioon, joka neuvoo käyttäjää kirjautumaan sisään
    b. .follow-nappiin liitetty kuuntelija vie kyseisen tietueen IMDb ID:n followFunction-funktioon
    c. followFunction syöttää imdbid:n savefollow.php-tiedostoon ja toiminnoksi “save”
    d. savefollow.php tarkistaa syötteen
    e. savefollow.php syöttää HT_followList-tauluun kyseisen tietueen ID:n ja käyttäjän ID:n merkiten, että käyttäjä seuraa kyseistä elokuvaa/tv-sarjaa

Jälkisanat

Palvelun toteuttaminen osoittautui yllättävän aikaavieväksi projektiksi - varsinkin yksin koko projektin toteuttaessa. Arvioisin, että aikaa kului n. 70 tuntia. Halusin kuitenkin tehdä välttämättä yksin koko projektin, oppiakseni mahdollisimman paljon. Kurssilla opin paljon itselleni pitkälti ennen tuntemattomasta javascriptistä (ajax/jquery) ja php:sta, ja harjoitustyön laajuudesta johtuen nämä kävivät hyvinkin tutuiksi. Mielenkiintoisimmaksi uutena opittuna asiana kurssilla koin Memcached-ominaisuuden, jota pitää käyttää suhteuttaen palvelimen muistin määrään ja vierailijamäärään. Joistain ominaisuuksista ja toiminnallisuuksista täytyi luopua puhtaan ajanpuutteen vuoksi, mutta luultavasti tulen vielä hiomaan palvelua itselleni sopivammaksi, jotta voin käyttää sitä itse aktiivisesti.