

Pokemon tietokanta

Nimi: Niko Novitsky

Opiskelijanumero: 014013656

Johdanto

Järjestelmän tavoitteena on luoda kattava tietokanta joka pitää sisällään kaikki (tällä hetkellä 719) Pokemon hahmoa ja niiden tärkeimmät perustiedot. Kuten nimi, tyyppi, evoluutiot, EV-arvot, mahdolliset liikkeet joita Pokemon voi oppia, sekä useita muita tietoja joita työn edetessä tulee tarpeelliseksi tai hydylliseksi tallettaa.

Järjestelmän toinen tavoite ja käyttötarkoitus, on auttaa käyttäjää rakentamaan ja luomaan oma Pokemon tiiminsä. Käyttäjä voi siis valita tietyt pokemonit ja tallettaa ne omalle tililleen näin luoden pokemon tiimin. Tiimiä on voitava tarkastella ja eri pokemonien tietoja oman tiimin sisällä on voitava myös muuttaa jotta koko järjestelmässä olisi mitään järkeä. Ylläpitäjä taas voi muuttaa yleisiä pokemon tietoja.

Käyttötapaukset

Järjestelmällä tulee olemaan kolme käyttäjäryhmää, rekisteröimätön käyttäjä, rekisteröity käyttäjä sekä ylläpitäjä.

- Pokemonien tarkastelu.
- Pokemonien etsintä tietyillä hakukriteereillä
- Pokemonien lisäys tietokantaan
- Pokemonien poisto tietokannasta
- Pokemonin lisäys joukkueeseen
- Pokemonin listäys uusin joukkoihin
- Pokemonin poisto joukkueesta
- Oman pokemonin tietojen muuttaminen
- Yleisen pokemonin tietojen muuttaminen
- Sivulle rekisteröityminen
- Sivulle kirjautuminen

Rekisteröimätön käyttäjä pystyy tarkastelemaan ja etsimään pokemoneja, sekä rekisteröitymään tietokantaan

Rekisteröity käyttäjä pystyy kirjautumaan sivulle, tarkastelemaan ja etsimään pokemoneja, käyttäjä pystyy myös liittämään pokemoneja tililleen aktiiviseen ryhmään sekä varastoon sekä muodostamaan uusia joukkueita pokemoneista, esimerkiksi vastustajan joukkueen. Käyttäjä pystyy muuttamaan pokemoniensa tietoja ja arvoja.

Ylläpitäjä pystyy lisäämään ja poistamaan pokemoneja tietokannasta sekä muuttamaan tietokantaan tallennettujen pokemonien arvoja.

Käyttötapauskuvaukset

Tapaus 1:

Rekisteröity käyttäjä haluaa etsiä ja tarkastella tietyn tyyppisiä pokemoneja. Hän avaa hakusivun jossa annetaan lista mahdollisista hakukriteereistä. Käyttäjä pystyy valitsemaan muunmuassa pokemonin tyyppin, evoluution vaiheen, tietyt stat arvot, nimen, jne.

Tapaus 2:

Käyttäjä haluaa muuttaa erään pokemoninsa hyökkäysarvoa. Käyttäjä navigoi tililleen ja valitsee halutun pokemonin, järjestelmä antaa käyttäjälle kyseisen pokemonin tiedot ja mahdollisuuden muuttaa niitä.

Tapaus 3:

Ylläpitäjä huomaa että uusi pokemon peli on tullu ja haluaa näin lisätä pokemoneja tietokantaan. Ylläpitäjä avaa admin osion ja valitsee kohdan mistä pokemoneja lisätään. uuden pokemonin tiedot kirjataan ylös, järjestelmä tarkistaa ne puutteiden jne varalta ja lisää ne tämän jälkeen järjestelmään.

Järjestelmän yleisrakenne

SQL kansio pitää sisällään tietokantojen luomiseen ja tuhoamiseen tarvittavat tiedostot. Add-test-data.sql ja add-required-data.sql ovat myös välttämättömiä ajaa järjestelmää asennettaessa

html-demo kansio pitää sisällään sivujen PHP koodin, eli kontrollerit. Nämä tiedostot huolehtivat oikeiden viestien ja argumenttien lähettämisestä objektien malleille tietokannan käyttöä varten

MyPage.php pitää sisällään funktiot pokemonien poistoa varten käyttäjiltä, sekä datan muuntamista varten

Search.php pitää sisällään funktion pokemonin lisäystä varten käyttäjälle

SignIn.php pitää sisällään käyttäjätilin luomisen, sekä etsinnän

LogOut.php pitää sisällään funktion käyttäjän loggaamiseen ulos järjestelmästä

Libs kansiota löytyy järjestelmän selkäranka.

Views kansiossa on sivujen ulkoasun määrittävä HTML koodi, CSS koodi taas löytyy CSS kansiota

Models kansio pitää sisällään objektit ja niiden käyttöön vaaittavat toiminnot.

Pokemon.php on perus pokemoneja hallitseva luokka mistä löytyy funktiot niin listaukseen, poistoon kuin perustietojen hakemiseen

move.php pitää sisällään tiedot pokemonien liikkeistä, niihin vaadittavat SQL kyselyt jne.

trainer_pokemon on hyvin samanlainen luokka kuin pokemon, kuitenkin kaikki toiminnallisuus sidotaan käyttäjiin ja heidän pokemonihinsa.

user.php on perus käyttäjää hallitseva luokka.

Lopuksi libs kansion juuresta löytyy ommon.php mistä löytyy yleisiä funktioita jotka hulehtivat käyttäjän kirjautumisesta ja loggaamisesta ulos

Tietokantayhteys.php huolehtii tietokantayhteydestä.

Lopuksi resource kansiota löytyy tiedon parsimiseen tarvittavat funktiot ja tiedostot. Parsemoves, parsepokemon aj parsetypes ovat kaikki nimensä puolesta yksiselitteisiä

Asennusohjeet

1. Aseta kaikki tiedostot nettiin näkyvään kansioon, esim htdocs.
2. Palauta kaikki data komennolla "psql dbname < db_with_data_dump"

Järjestelmä on nyt valmis käytettäväksi.

Mikäli data halutaan rakentaa uudestaan:

1. Aseta kaikki tiedostot nettiin näkyvään kansioon, esim htdocs.
2. suorita create-tables.sql tiedosto jolloin tietokanta pystytetään

3. Resource kansista löytyy kolme php tiedostoa, ParseMoves, ParsePokemon ja ParseTypes.
4. Suorita ensin ParsePokemon joka hakee ja lisää kaikkien POkemonien datan tietokantaan
5. Suorita add-test-data.sql joka pitää sisällään tyyppien perustiedot
6. SEuraavaksi suoritetaan resource kansista PARseMoves ja PARseTypes
7. Viimeiseksi add-required-data.sql
8. Järjestelmän pitäisi olla nyt käyttövalmis.

Testausohjeet:

Järjestelmän etusivu löytyy [tästä](<http://nikonovi.users.cs.helsinki.fi/PokeDB/html-demo/Etusivu.php>)

Millä tahansa tunnoksella voi rekisteröityä järjestelmään, testaustunnuksena olen itse käyttänyt:
kayttaja - salasana

admin salasana: secret_admin - superpassword

Adminilä on oikeus muuttaa globaalin Pokemon tietokannan arvoja.

Bugit, oiminnallisuus ja kehitys

Suurmpia puutteita järjestelmässä:

- Hakutoiminnon puute
- Lajittelun puute
- Admin ei pysty lisäämään pokemoneja vielä

Bugeja:

- Kirjautumissivu näyttää aina vihreilmoituksen
 - Ulkoasu on hiukan omituisesti skaalautuva
 - Hidas toiminta mobiililaitteilla
-

Omat kokemukset:

Järjestelmän tekeminen itsessään tuntui melko hauskalta ja yllättävän simppeliltä, enemmän jouduin taistelemaan PHP:n ja HTML:n keskustelun kanssa. useiden muutjien ja argumenttien passaaminen HTMLstä PHP:hen oli hiukan kinkkkistä, tähän olii voinut auttaa AJAX ja javascriptin yleinen käyttö, mutta yritin välttää tätä.

Toienn hiukan päänsärkyä aiheuttava asia oli SQL lauseiden kanssa taistelu jotta kaikki tarpeellienn saadaan tietokannasta ulos

Itse projektin kannalta suurin henkilökohtainen ongelmani oliviikkopalautukset jotka kaikki menivät lähes täysin metsään pienten virheiden ja hajamielisyyden takia. Unohdin lisätä käyttäjätunnukset testausa varten, muokkasin väärää readme tiedostoa, jne.