

Inlämningsuppgift 2 (U/G)

The Internet Dog Database (IDDb)

Ni ska bygga en webbapplikation där användare kan publicera och administrera information om sina hundar samt ta del av andra användares hundar. Sidan ska byggas med hjälp av PHP, HTML och CSS - JavaScript är inte tillåtet att använda.

Inlämning

Uppgiften ska lämnas in som en zip-fil med alla nödvändiga filer (PHP, HTML och CSS) på Canvas. Glöm inte att gå igenom hela uppgiftsbeskrivningen (Sidor, Databas, Övriga krav) innan inlämning för att minska risken för fel. Vi kommer att starta en lokal webserver med hjälp av PHP när vi granskar uppgiften, då upptäcker vi snabbt om någonting inte fungerar som det ska.

Beskrivning

Er webbapplikation ska bestå av diverse olika sidor (se listan nedan). Det finns inga krav på en speciell design utan webbapplikationen ska vara godtagbart responsiv och användarvänlig. Kom ihåg att detta ska göras med HTML och CSS, **inte** JavaScript. Längre ner finner ni information om upplägget på er databas samt övriga krav på funktioner som ska finnas utöver det som listas under "Sidor".

För att få en bättre idé över upplägget finner ni även en videoinspelning av ett lösningsförslag i denna mapp. Samt en mapp ("Exempel") innehållande ett förslag på filstrukturen, vilket kan underlätta att komma igång. Ni bestämmer själva hur ni vill lägga upp strukturen för era filer, förslag på filnamn som finns i listan nedan baseras på dom i mappen "Exempel".

Sidor

- A. "**Home**" [index.php]: Kort välkomsttext samt länkar till "Sign in" och "Dogs".
- B. "**Sign in**" [sign-in.php]: Innehåller ett formulär för att logga in med email och lösenord (formuläret kan antingen skicka sin data till en separat fil eller till sig själv). Skickar användaren till "Dogs" vid inloggning, annars ska ett relevant felmeddelande visas. När en användare har loggat in ska ni spara dennes "id" i er session (så ni har det tillgodo).

- C. **"Dogs"** [list.php]: Innehåller en lista över alla hundar i databasen. Det ska finnas möjlighet att kunna (1) klicka på en hunds namn (då skickas dom till sidan "Show") samt (2) kunna klicka på hundrasen för att filtrera alla hundar därefter (detta ska styras genom GET-parametern "breed", t.ex. "list.php?breed=husky").
- D. **"Show"** [show.php]: Innehåller information om en specifik hund. Detta ska styras genom GET-parametern "id" (t.ex. "show.php?id=3"). Om hunden inte finns (dvs. "id" inte finns) så ska ett relevant felmeddelande visas.
- E. **"Add"** [add.php]: Innehåller ett formulär för att lägga till en ny hund (formuläret kan antingen skicka sin data till en separat fil eller till sig själv). När hunden är tillagd ska ett relevant meddelande visas, annars ska ett relevant felmeddelande visas. Nyttja användarens "id" som ni sparar i er session för att ange vem hundens "owner" är.
- F. **"Profile"** [profile.php]: Innehåller en lista över en användares alla hundar. Namn och hundras behöver inte vara länkar i denna listan (som på sidan "Show"). Det ska däremot finnas en länk för varje hund som gör att hunden raderas från listan (t.ex. genom en länk till "delete.php?id=3").
- G. **"Delete"** [delete.php]: Detta ska inte vara en sida. Ni ska genom denna fil radera en hund (detta ska styras genom GET-parametern "id") och sedan skicka tillbaka användaren till sidan "Profile".
- H. **"Sign out"** [sign-out.php]: Detta behöver inte vara en sida, det går bra att direkt skicka användarna vidare till t.ex. "Sign in" eller "Home".

Beroende på om användaren är inloggad eller inte ska olika länkar synas i navigationen. Inte inloggad: "Home - Dogs - Sign in", inloggad: "Home - Dogs - Add - Profile - Sign out".

Databas

Er databas ska lagras i en JSON-fil, t.ex. "db.json". Den ska innehålla två typer av data: hundar (dogs) och användare (users). Eftersom det inte finns någon registreringssida kommer vi att tillhandahålla en exempelfil över databasen, så ni har någonting att utgå från (denna hittar ni i mappen "Exempel"). Det innebär att om ni vill lägga till en ny användare måste ni manuellt redigera er databasfil. När ni konverterar till JSON kan ni lägga till en extra parameter för att få filen läsbar (indenterad), vilket underlättar när ni sedan manuellt redigerar filen:

```
json_encode($data, JSON_PRETTY_PRINT);
```

En hund ska ha följande nycklar: *id, name, breed, age, notes, owner*. Nyckeln "id" ska öka med 1 för varje ny hund som läggs till. Nyckel "owner" ska vara den användares "id" som precis skapade hunden.

En användare ska ha följande nycklar: *id, email, username, password*. Nyckeln "id" ska öka med 1 för varje ny användare som läggs till. Nyckeln "username" är den som används i listan över alla hundar (ägarens namn). Nyckeln "email" och "password" är den kombination som används vid inloggning.

Övriga krav

- Om en användare besöker sidorna "Add", "Profile" eller "Delete" utan att vara inloggad ska dom skickas till sidan "Sign in".
- Om en användare fyller i något formulär på ett felaktigt vis, eller att en inloggning misslyckas, så ska dom få ett relevant felmeddelande.
- Det enda kravet på samtliga formulär är att dom inte får vara tomma. Det är därför valfritt att kontrollera mer saker som att en email innehåller "@" eller att ålder på en hund är någorlunda rimlig (t.ex 0-16).
- Det ska finnas en huvudnavigation som är konstant synlig och lätt att navigera med.
- Vid inlämning ska er databasfil ha en mindre mängd innehåll för att underlätta vår testning och rättning. Ha därför minst tre användare med två hundar var.
- **All er kod** (PHP, HTML och CSS) **ska ha god formatering** (indentering, radbryt, m.m.) **och dokumentation** (kommentarer samt relevanta namn på era variabler och funktioner).