Databanken

Voorkennis en uitwerking

Deze opgave bouwt verder op een gegeven, onafgewerkte, simpele todolist-website (terug te vinden op Toledo). In te bouwen functionaliteiten zijn het communiceren met een MySQL databank (zowel lezen als schrijven), en het afhandelen van de resultaten. Vergeet niet om de nodige beveiligingsmaatregelen te voorzien. **Beveilig je pagina tegen XSS, SQL Injectie, en Parameter Tampering.** Vergeet tenslotte niet om foutmeldingen te onderdrukken.

Het is sterk aangeraden alle code en de theoretische achtergrond die dit project vereist grondig te bestuderen alvorens je naar het labo komt. Bestudeer eveneens de reeds voorziene code. Je zal merken dat heel wat in te bouwen zaken reeds in commentaar uitgewerkt zijn. De opgaven moeten in volgorde afgewerkt worden.

Het huishoudelijk reglement is zoals steeds van toepassing.

Timing en upload

Dit labo loopt over één sessie (3u) en dient aan het einde van dit labo via Toledo geüpload te worden. Thuis werk je het labo verder af. Upload één .zip bestand met daarin je .php bestanden.

Doelstellingen

- Kunnen communiceren met een MySQL-databank vanuit PHP via de PDO extensie
- Prepared statements met PDO kunnen implementeren
- De resultaten van een prepared statement verwerken
- Kunnen opbouwen van logische code, op basis van reeds bestaande voorbeeldcode

Op schijf

- De locatie waar dit project opgeslagen wordt is <webroot>/voornaam.achternaam/labo04
- Bewerk de reeds voorziene bestanden

Troubleshooting phpMyAdmin

De instellingen van phpMyAdmin bevinden zich in de config-file met als naam config.inc.php. (Locatie op MAMP: `/Applications/MAMP/bin/phpMyAdmin/config.inc.php`)

- Bij de foutboodschap 'Wrong permissions on configuration file ...' ontzeg je de 'others' schrijfrechten opt bestand (`su - cisco ; sudo chmod o-w /Applications/MAMP/bin/phpMyAdmin/config.inc.php`)
- Zorg ervoor dat phpMyAdmin je steeds vraagt om in te loggen. Hiervoor zet je in config.inc.php het auth-type op 'cookie'. Dit bespaart je niet alleen gedoe bij de wijziging van een paswoord ('permission denied'); het is gewoon ook veel veiliger (!).

Opgelet !!!

Vergeet niet om alles te beveiligen tegen XSS en SQL Injection:

- Alles wat in een database/query gestopt wordt: via een prepared statement
- Alles wat op het scherm weergegeven wordt: `htmlentities()` gebruiken

Opgaves

1. Importeer de gegeven SQL dump via phpMyAdmin en wijzig de charset van zowel de databank als de tabellen naar utf8mb4.
2. Voorzie de functie showDbError() in het bestand functions.php, alsook het vanuit die functie naartoe verwezen error.php bestand. Breid de functie zodanig uit dat er op basis van een zelf te definiëren constante DEBUG de foutmelding al dan niet meteen op het scherm weergegeven wordt.
   - Staat DEBUG op true, dan worden fouten op het scherm weergegeven (voor jou, als developer). Deze instelling behoud je tijdens het developen.
   - Staat DEBUG op false, dan worden fouten zoals standaard voorzien gelogd (voor jou, als developer) en krijgt de bezoeker een mooie foutpagina te zien. Deze instelling gebruik je op de gepubliceerde versie van de site.
3. Het bestand browse.php omvat een basis en is voorzien van heel wat commentaar. De commentaar geeft de door jou te implementeren logica weer:
   - Het opbouwen van de connectie naar de MySQL-databank, met gebruik van een configuratiebestand
   - Het ophalen van de todo-items

- Het overlopen en weergeven van de todo-items
- Het afhandelen van het verzendformulier zodat men items kan toevoegen

Wanneer we doorlinken naar edit.php en delete.php , geven we de id mee in de URL.

- De array $formErrors krijgt een foutboodschap mee wanneer de gebruiker het tekstveld leeg laat of wanneer de gebruiker een niet-bestaande prioriteit probeert te 'tweaken'. Dit soort foutboodschappen kan je gewoon kwijt in het HTML-gedeelte in een element.

4. Het bestand edit.php werd eveneens voorzien van commentaar met instructies. Werk ook dit basisbestand verder uit.

Gebruik een hidden field `moduleAction` om na te gaan of de form verstuurd werd. Gebruik ook een hidden field met de id van het te bewerken bestand.

5. Het bestand delete.php is nog leeg. De opbouw is echter analoog met hetgeen in edit.php gebeurt. Baseer je op dit laatstvermelde bestand om zelf delete.php op te bouwen. Je zal merken dat het uiteindelijke bestand niet zo heel veel met edit.php verschilt.