

13.04.2010

A stylized question mark logo is centered on the page. The question mark is white with a blue outline and is set against a black background. From the top and bottom of the question mark, a series of blue and white rays radiate outwards, resembling a sunburst or a starburst. The entire logo is contained within a square frame with a black border. A horizontal grey bar with a black section on the left side overlaps the middle of the logo, and the text 'DOKUMENTATION MYPHP-QUIZ' is written in white capital letters across this bar.

DOKUMENTATION MYPHP-QUIZ

## Worum geht's?

Ziel des Projektes myPHP-Quiz ist die Umsetzung eines einfachen Quizsystems in PHP5. Es soll dabei bestimmten Benutzern (sog. Quizmastern) möglich sein, Quizsysteme anzulegen. Ein Quizsystem besteht aus einer Reihe von Fragen, welche entweder zufällig oder in einer bestimmten Reihenfolge aufgeführt werden, und den zu den Fragen möglichen Antworten. Es ist immer mindestens eine Antwort richtig – es können auch mehrere Antworten korrekt sind. Insgesamt kann es im System 3 verschiedene Arten von Antworten geben:

- RadioButtons: Hier ist per Definition nur eine Antwort korrekt.
- CheckBoxen: Hier sind 1 bis  $n$  (Anzahl der Antwortmöglichkeiten) Antworten korrekt. Der Benutzer erhält nur bei Auswahl ALLER richtiger Antworten einen Punkt (Frage korrekt beantwortet).
- TextBox: Hier gibt es offensichtlich nur eine Antwortmöglichkeit (in Form einer TextBox). Es sind aber  $n$  (Anzahl der im System eingegebenen Antworten) korrekt. Zur korrekten Beantwortung der Frage wird (per Definition) nur die Übereinstimmung mit einer vorgegebenen Antwort benötigt.

Am Ende eines Quiz wird dem Benutzer je nach Einstellung des Quizsystems über folgende Möglichkeiten informiert:

- Seine korrekten Antworten / der korrekte Anteil in Prozent.
- Die Möglichkeit sich zu registrieren um an der Verlosung teilzunehmen.
- Ein weiteres Quiz zu spielen.

Entschließt sich der Benutzer zum Anmelden am Gewinnspiel (falls dies eingestellt wurde), so kann der zuständige Quizmaster dieses Quiz bzw. ein (normalerweise nur einer, also: der) Superadmin ihn als Teilnehmer an dem jeweiligen Quiz sehen. Es besteht nun die Möglichkeit, dass der Benutzer bei einer – vom zuständigen Quizmaster angestoßenen – Verlosung gezogen wird und somit per E-Mail über seinen Gewinn benachrichtigt wird.

## Umsetzung in PHP

Es wurden prinzipiell zwei Webseiten angelegt:

- admin.php (\*)
- index.php (\*\*)

Während man sich in (\*) direkt mit einem Adminaccount einloggen muss, kann man (\*\*) auch als normaler Benutzer ohne einen Account verwenden. In (\*) werden neue Quiz angelegt, angelegte verändert (Daten geändert, Fragen und Antworten modifiziert) und Quizmaster / Teilnehmer verwaltet. In (\*\*) können sich Benutzer registrieren oder an Gewinnspielen teilnehmen. Dazu muss vorher das Quiz, an dem man teilnehmen möchte, gespielt worden sein.

Die Besonderheit des Systems besteht darin, dass alles auf Sessions aufbaut. Hat ein Benutzer keine Cookies aktiviert, kann er nicht an einem myPHP-Quiz teilnehmen. Durch die Speicherung der Daten in Cookies wird die Manipulation des Systems deutlich schwieriger als bei Speicherung der Daten im Formularquelltext der Webseite.

Neben den zwei angegebenen Webseiten gibt es noch jede Menge weiterer \*.php-Dateien, welche von den zwei Webseiten genutzt werden um den Code übersichtlich zu halten, und bereits erstellten Code wiederverwenden zu können.

## Aufgaben der einzelnen Dateien

---

Verzeichnis „admin“: Hier liegen alle für die Adminwebseite „admin.php“ erforderlichen Dateien.

- „addquiz.php“ beinhaltet den Code zum Hinzufügen und Verändern von Quizdaten.
- „home.php“ beinhaltet den Code zur grundsätzlichen Anzeige der vorhandenen Quiz.
- „participants.php“ erlaubt dem Quizmastern die Teilnehmer ihrer Quiz anzusehen und die Verlosung der Preise zu starten.
- „qm.php“ erlaubt Superadmins die Veränderung der Quizmasterdaten.
- „questions.php“ beinhaltet das Interface um Fragen zu bestehenden Quiz hinzuzufügen oder zu verändern. Das Verändern von bestehenden Fragen beinhaltet das Hinzufügen oder Ändern von Antworten.
- „user\_a.php“ ist die Klasse, welche in der Session gespeichert wird – sie enthält den Adminuser und die notwendigen Zugriffsfunktionen.
- „vars.php“ wird von Superadmins benutzt um die Systemvariablen zu verändern.

Verzeichnis „core“: Hier liegen alle für die Webseite „index.php“ erforderlichen Dateien.

- „functions.php“ ist eine Sammlung von nützlichen Funktionen für alle Webseiten des myPHP-Quiz-Projektes. Neben Funktionen zum generieren von Passwörtern und überprüfen dieser, ist eine Funktion zum Überprüfen der E-Mail Adresse, eine Funktion zum Verhindern von SQL Injections und die Funktion zum berechnen des Quiz-Ergebnisses enthalten.
- „join.php“ wird aufgerufen wenn der Benutzer einem Gewinnspiel beitreten will.
- „login.php“ regelt den Login / Logout des Benutzers.
- „quizmain.php“ stellt das Herz des Projektes dar. Hier wird der eigentliche Quizablauf geregelt.
- „register.php“ enthält das Benutzerprofil und die Anmeldemaske inkl. Logik.
- „title.php“ beinhaltet die Logik zur Auswahl des Webseitentitels.
- „user\_n.php“ beinhaltet die Klasse, welche die nötigen Funktionen zum Überprüfen von Benutzern enthält.

Das Verzeichnis „DB“ enthält die Datei „connect.php“, welche den Datenbankzugriff ermöglicht.

Die eben genannten 3 Verzeichnisse müssen über facl's geschützt werden (von außen nicht zugreifbar, nur intern Zugriff möglich).

Das Verzeichnis „img“, sowie die Dateien „quiz.css“, „quiz.ico“ und „quiz.js“ sind nur zur Anzeige bestimmt, müssen von außen lesbar sein und können nach Belieben geändert werden.

## Noch nicht eingebaute Features

---

Der aktuelle Projektstand (13.04.2010) repräsentiert nur eine kleine Zahl an Möglichkeiten. Im System sind folgende Punkte noch relativ leicht zu integrieren:

- Benutzer können Ihre Daten selbst ändern: Diese müssten dazu auf Konsistenz geprüft werden, oder ein Admin müsste über die Änderung informiert werden. Wenn der Admin bei der anschließenden Prüfung auffällige Daten feststellt, könnte er den Benutzer sperren.
- Admins können die Passwörter der Benutzer zurücksetzen – wenn dies von ihnen angefordert wurde, und über einen Bestätigungslink in ihrer E-Mail bestätigt wurde.
- Die Preise werden in eine extra Tabelle ausgelagert mit einem kombinierten Primärschlüssel bestehend aus dem Preisrang (1. Preis, 2. Preis, 3. Preis,... also: 1, 2, 3, ...) und der Quiz ID. Neben dem Primärschlüssel wären noch ein Preistext und eine Preisanzahl vorhanden, so kann der 1. Preis 5x vergeben werden, der 2. Preis 10x etc. (Beispielhaft).
- Es würde noch jede Menge mehr Optionen geben, z.B. das Design betreffend (Sidebar, News, AGB, ...) und mehr Optionen für das Quiz, z.B. auch eine zufällige Anordnung der Quizantworten betreffend (gilt nur für Radiobuttons und Checkboxes).

## Installation des Systems

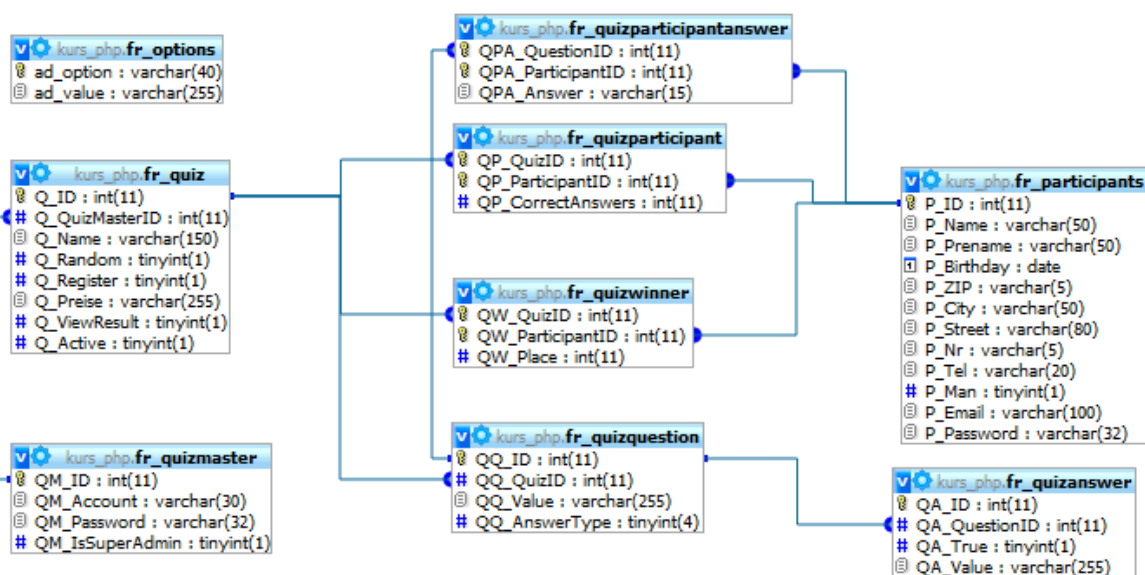
Die Installation von myPHP-Quiz erfolgt auf einem Webserver, welcher über mindestens über PHP 5.1 und MySQL 5 verfügt. Über die Datei „dbqueries.sql“ kann man sämtliche Datenbanktabellen erstellen lassen. Wichtige Kenndaten sind:

- Datenbankname: kurs\_php
- Datenbankbenutzer: kurs\_php
- Benutzerpasswort: (wird immer im php/MySQL Kurs bekannt gegeben)
- Administratoraccount für das Quizsystem: Admin
- Administratorkennwort für das Quizsystem: admin (Standardadmin sollte nach Anlegen eines neuen (Superadmin-) Accounts gelöscht werden)

Das Administratorkennwort kann nach der Installation geändert werden.

## Datenbankentwurf

Die folgende Abbildung zeigt die Datenbankrelationen von myPHP-Quiz.



Die Tabelle „FR\_Options“ wurde erstellt, um das System sehr variabel zu halten. Hier können allgemeine Variablen erstellt, und deren Werte gespeichert werden. Alle anderen Tabellen sind zur Speicherung bzw. Ausgabe der Quizdaten erforderlich. Einige Besonderheiten der Datenbank sollen im Folgenden erklärt werden:

- In der Tabelle „FR\_QuizParticipant“ wird grundlegend gespeichert ob ein Teilnehmer an einem Quiz teilgenommen hat, und falls dem so, wie viel korrekte Antworten der Teilnehmer abgegeben hat. Obwohl man über die Tabelle „FR\_QuizParticipantAnswer“ auch die Anzahl der korrekten Antworten ablesen könnte, ist dies so sinnvoll, da ein Quiz ja nachträglich verändert werden kann, und neue Antworten in einem solchen Fall dazu kämen oder alte wegfallen könnten. Außerdem spart man über diesen Weg einiges an Berechnungen.
- In der Tabelle „FR\_QuizParticipantAnswer“ wird die Antwort in einem Zeichen-Wert gespeichert – statt als Referenz. Das hängt damit zusammen, dass man bei CheckBoxen (AnswerType = 2) auch mehrere Antworten angeben kann und man diese über eine Zwischentabelle (m:n) speichern müsste. In diesem Fall wird anstelle der m:n Beziehung ein String verwendet, welcher über die PHP-Funktion implode (Array in String umwandeln) mit Daten gefüttert wird.
- Eine zentrale Tabelle ist die „FR\_QuizMaster“-Tabelle. Nicht nur, dass man mindestens einen QuizMaster-Account benötigt um sich einloggen und die Funktionalität der „admin.php“-Datei nutzen zu können, sondern auch Datenbanktechnisch ist ein Quizmaster Account zum Anlegen von Quiz notwendig. Jedes Quiz benötigt einen Quizmaster (Verantwortlichen), jede Quizfrage benötigt ein Quiz und jede mögliche Quizantwort benötigt eine Quizfrage zu der die Antwort gehört.

### Abschließende Bemerkung

Sicherheit ist alles, weshalb die Sicherung der Unterverzeichnisse (v.a. des DB-Verzeichnisses) über entsprechende facl's notwendig und sicher ist. SQL Injections sollten mit dem aktuellen Code unterbunden sein und Benutzereingaben sicher behandelt werden.

Das Projekt sollte das im Kurs gelernte umsetzen und durch neue, gute Konzepte (E-Mailadressen über reguläre Ausdrücke überprüfen, Optionen in Hashtable speichern, Instanzen von Klassen in Sessions speichern) erweitern.