

# Php-Quiz

SEBASTIAN PEHLKE 2285904

FREDERIK MERTEN 2288669

MAXIMILIAN KRAFT 2297992

## Inhalt

1 Aufgabe der Software.....	2
2 Installationsanleitung .....	2
3 Bedienungsanleitung .....	3
4 Systemarchitektur .....	9
5 Beschreibung eines technischen Teilaspektes .....	9
6 Fazit, Auswertung .....	10

## 1 Aufgabe der Software

Die vorliegende Software ist ein Quiz, das mit php, html und mysql realisiert wurde. Der Nutzer kann sich auf der Startseite einloggen oder registrieren. Wählt er die Registrierung aus, muss er ein Formular mit Nutzernamen und Passwort ausfüllen. Die Daten werden in der Datenbank gespeichert. Das Einloggen auf der Startseite ist damit möglich.

Loggt ein Nutzer sich ein, wird eine Session gestartet. Es wird angezeigt wie der Nutzer heißt, welchen Rang der Nutzer hat, seinen aktuellen Score und die nächste zu beantwortende Frage.

Es gibt zwei Ränge: Nutzer und Admin. Dem Nutzer ist es nur gestattet Fragen zu beantworten, währenddessen der Admin auch neue Fragen eintragen kann.

Der Score wird mit dem Benutzernamen, dem Passwort, der Benutzer-ID und dem booleschen Wert, ob der Nutzer Admin ist oder nicht, gespeichert.

Die Fragen werden zufällig aus der Datenbank ausgewählt.

Nutzer können so viele Fragen beantworten wie sie mögen.

## 2 Installationsanleitung

Damit die Software laufen kann, muss ein Server eingerichtet werden und eine Datenbank zur Verfügung stehen.

Die Quelldateien müssen auf den Server geladen werden.

In der Datei database.php sind Variablen einzutragen, damit eine Verbindung zur Datenbank hergestellt werden kann. Die Variablen beinhalten: Servername, Username, Passwort, Name der Datenbank.

In der Datenbank müssen zwei Tabellen erstellt werden. Eine Tabelle mit dem Namen 'fragen' und eine mit dem Namen 'users'.

Die Tabelle 'fragen' hat 5 Spalten.

- id, fängt bei 0 an und erhöht sich mit jedem Eintrag automatisch

- frage, beinhaltet die ausgeschriebene Frage
- antwort1, eine Anreihung von Chars
- antwort2, eine Anreihung von Chars
- richtig, gibt die Richtige Antwort an (1 oder 2)

Die Tabelle 'users' hat 5 Spalten.

- name, der bei der Registrierung eingegebene Nutzername
- password, das gehashte Passwort
- score, die aktuelle Score
- admin, bei 1 ist der Nutzer Admin und kann Fragen eintragen
- id, fängt bei 0 an und erhöht sich mit jedem Eintrag automatisch

### 3 Bedienungsanleitung

Die Software startet auf der Seite index.php.



Hier wird der Nutzer geben sich einzuloggen oder sich zu registrieren.

Hat der Nutzer schon einen Account, kann er sich einloggen. Hat er keinen Account sollte er sich registrieren.

# Registrieren!

oder [hier einloggen](#)

Name:

Passwort:

Passwort:

Daten absenden

Name und Passwort können beliebig gewählt werden. Beide Passwörter müssen jedoch übereinstimmen. Ist der Nutzer erfolgreich erstellt, kann der Benutzer sich einloggen.

# Einloggen!

oder [hier registrieren](#)

Name:

Passwort:

Daten absenden

Das Einloggen ist nur möglich, wenn ein passender Datenbankeintrag vorliegt.

## Quiz

Willkommen Paul  
Eingeloggt als: Nutzer

Score: 0

Anzahl Fragen in der Datenbank: 5

Frage 5

**Wie nennt man die Schwester der Mutter?**

Der Nutzer sieht jetzt einige Informationen über seinen Account und der Datenbank. Er kann direkt anfangen Fragen zu beantworten. Nach dem Beantworten wird der Nutzer weitergeleitet. Er sieht, ob die Antwort richtig war und seinen aktuellen Score. Mit klicken auf Weiter wird er wieder auf die zu vorige Seite geleitet.

Quiz
<b>Ergebnis:</b>
Richtig!
Score: 3
<a href="#">Weiter</a>

Der Nutzer kann sich jederzeit oben rechts in der Ecke der Seite ausloggen. Oder mit klicken der Überschrift 'Quiz' auf die Startseite zurückgelangen.

Ist der eingeloggte Nutzer ein Admin, so kann er weitere Fragen eintragen.



## Frage erstellen!

oder [zurück](#)

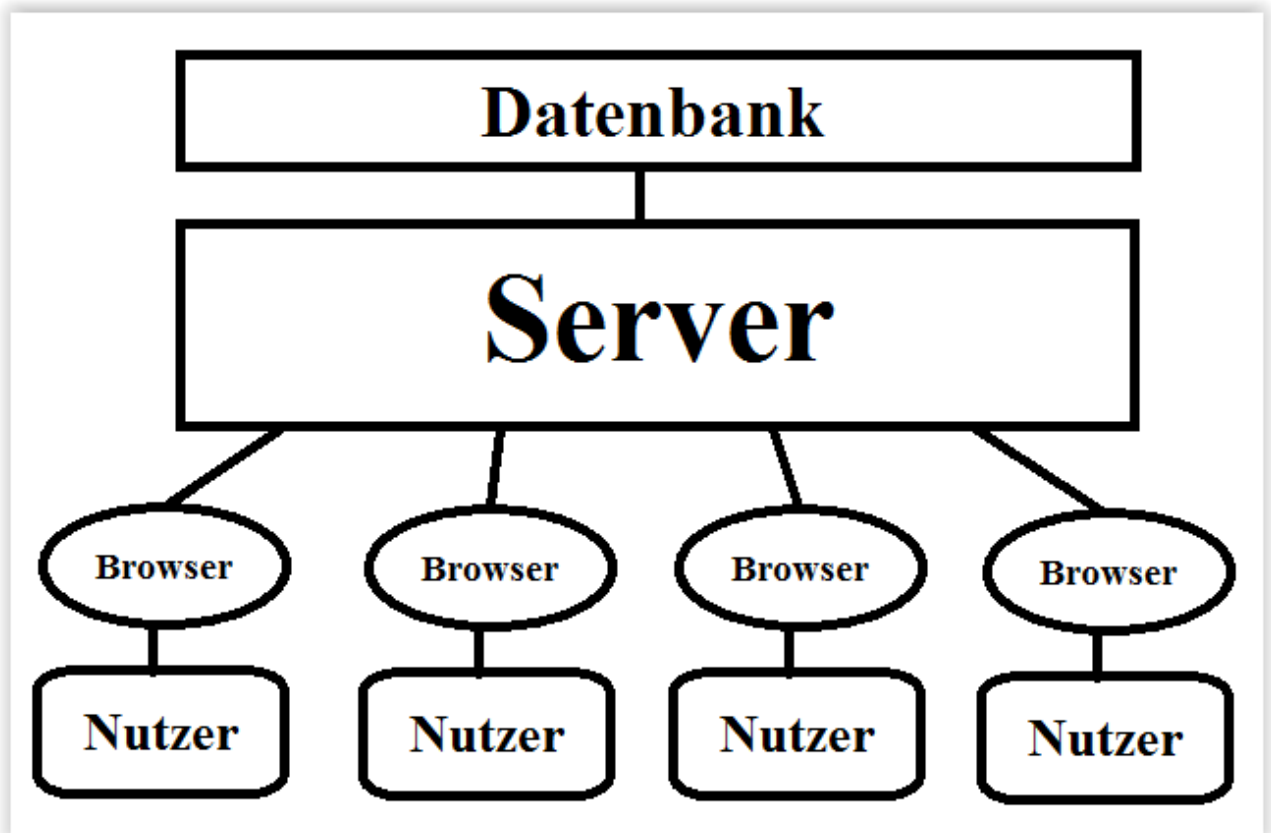
Richtige Antwort:

☐ 1 ☐ 2

Daten absenden

Die eingetragene Frage wird in der Datenbank gespeichert und ist in Zukunft für andere Benutzer sichtbar.

## 4 Systemarchitektur



Beliebig viele Nutzer können das Programm über einen Internet Browser aufrufen. Der Browser kommuniziert mit dem Server und der Server mit der Datenbank.

## 5 Beschreibung eines technischen Teilaspektes

Die Fragen werden mit Hilfe einer SQL-Methode aus der Datenbank geladen.

```
$records2 = $conn->prepare('SELECT id, frage, antwort1, antwort2, richtig FROM fragen WHERE id = :id');
```

Um die Fragen zufällig anzuzeigen, wird zunächst mit einer weiteren SQL-Methode die Anzahl der Reihen der 'fragen' Tabelle ermittelt.

```
$statement = $conn->prepare ("SELECT COUNT(*) AS anzahl FROM fragen");
```

Die Anzahl wird in einer Variable gespeichert. Mit der Methode rand() wird nun eine zufällige Zahl zwischen 0 und Anzahl der Reihen – 1 ermittelt. Es ist wichtig bei der Obergrenze 1 abzuziehen, da die SQL-Methode die Anzahl der Reihen ausgibt, wir aber

bei der Frage-ID bei 0 anfangen zu zählen. Bei 5 Fragen hat die erste Frage also die ID 0 und die 5. Frage die ID 4.

Die aus der rand()-Methode erhaltene Zahl wird als Parameter für die aktuelle Frage-ID übergeben.

```
$records2->bindParam('/:id', $_SESSION ["idR"]);
```

Zuletzt wird das Statement ausgeführt und in einem Array gespeichert.

```
$records2->execute();
```

```
$results2-> $records2->fetch(PDO::FETCH_ASSOC);
```

## 6 Fazit, Auswertung

Ein großes Problem ist, dass Nutzer bei richtig beantworteter Frage, die Seite mehrmals neu laden können. Das Formular wird somit erneut abgeschickt und der Score erhöht sich. Recherchen ergaben, dass dieses Problem nur mit JavaScript umgangen werden kann. In diesem Projekt sollte jedoch auf JavaScript verzichtet werden.

Des Weiteren sollten weitere Kontrollstrukturen eingefügt werden, die zum Beispiel sicherstellen, dass jeder Benutzername nur einmal existiert.

Außerdem ist die Anwendung nicht vor Hackangriffen, wie z.B. einer SQL-Injection, geschützt.

Fortführend könnte man noch eine Highscore-Tabelle, weitere Nutzerränge und mehr Antwortmöglichkeiten hinzufügen.