

Aufgabenstellung

PHP & MySQL

1 Allgemeines

Die Dauer der Prüfung liegt bei 5 Trainingseinheiten á 45min, wodurch sich eine Prüfungszeit von 3.75 Stunden ergibt. Dazwischen kann bei Bedarf jederzeit vom Teilnehmer eine Pause gemacht werden.

Wählen Sie eines der Beispiele. Sie können zwar während der Prüfung auf ein anderes Beispiel wechseln, jedoch kann nur ein Beispiel gewertet werden (dasjenige, das Sie abgeben).

Es sind sämtliche Unterlagen erlaubt, ebenso der Zugriff auf das Internet. Ein gegenseitiges Helfen ist untersagt, ebenso der Zugriff auf Beispiele anderer Prüfungskandidaten sowie selbst vorbereitete Beispiele. Die theoretischen Fragen können Sie optional beantworten, wobei diese Fragen weitere 10% einbringen können – Sie können demnach 110% erreichen. Sie müssen mind. 50% erreichen, um positiv zu sein.

Bitte legen Sie die Daten am Ende der Prüfung im Ordner ABGABEN (bzw. wie anderweitig vereinbart) gezippt in einer Datei namens VORNAME_NACHNAME.zip (also zB. Uwe_Mutz.zip) ab. Nachträglich eingebrachte Dateien können leider nicht berücksichtigt werden. Bitte beachten Sie insbesondere im Fall des datenbankbasierten Beispiels, dass Sie den Datenbank-Dump mit in die ZIP-Datei geben.

Bitte bedenken Sie, dass Sie keinen Wert auf grafische Ausgestaltung legen müssen. Es wird lediglich die Funktionalität und nicht das Design beurteilt.

2 Beispiele

2.1 Beispiel 1

Erstellen Sie ein Programm, wo sich ein User auf einer Seite namens *index.php* einloggen kann und danach eine Dateistruktur durchsuchen und ordnen können soll. Es soll in diesem Beispiel lediglich einen einzigen User geben, dessen Zugangsdaten wie folgt definiert sind:

- Username: admin
- Passwort: test123

Nach erfolgreichem Login soll dem User die Dateistruktur ausgehend von dem Verzeichnis **daten/** inkl. aller Unterverzeichnisse und Dateien ausgegeben werden. Bitte verwenden Sie

die Ihnen bereitgestellte ZIP-Datei, um die Datei- und Verzeichnisstruktur aufzubauen. Bei Klick auf einen Button »sortieren« sollen alle darin befindlichen Dateien in den dafür vorgesehenen Verzeichnissen abgelegt werden (die Sortierung soll auf Basis der Datei-Endung erfolgen):

- wird in irgendeinem der Verzeichnisse ein Textdokument (Endungen .txt, .docx, .rtf) gefunden, so soll dieses Textdokument in das Verzeichnis **sortiert/textdokumente** verschoben werden
- wird eine Bilddatei (Endungen .gif, .jpeg, .jpg, .png, .webp) gefunden, so soll diese in das Verzeichnis **sortiert/bilder** verschoben werden
- wird eine Präsentationsdatei (Endungen .pptx, .key) gefunden, so soll diese in das Verzeichnis **sortiert/presentationen** verschoben werden
- wird eine anderweitige Datei gefunden, so soll diese in das Verzeichnis **unsortiert** verschoben werden

Bitte achten Sie darauf, dass die Verzeichnisse **sortiert/textdokumente**, **sortiert/bilder**, **sortiert/presentationen** und **unsortiert** nur dann (mittels eines geeigneten PHP-Befehls) angelegt werden, wenn diese auch benötigt werden: mit anderen Worten soll beispielsweise das Verzeichnis **sortiert/bilder** nicht angelegt werden, wenn in der gesamten Verzeichnisstruktur kein Bild gefunden wurde.

Beachten Sie, dass bereits bestehende Dateien nicht überschrieben werden dürfen. Sollte eine Datei im Zielverzeichnis bereits existieren, so soll die zu verschiebende Datei im Dateinamen den Zusatz »_n« (wobei n für eine Zahl beginnend mit 1 steht) erhalten. Beispiele:

1. Es soll die Datei *bild1.jpg* nach **sortiert/bilder** verschoben werden. Dort existiert aber bereits eine Datei namens *bild1.jpg*
 1. Sollte noch keine Datei namens *bild1_1.jpg* existieren, so soll die zu verschiebende Datei unter dem Namen *bild1_1.jpg* im Zielverzeichnis (**sortiert/bilder**) abgelegt werden.
 2. Sollte bereits eine Datei namens *bild1_1.jpg* existieren, so soll die Datei unter *bild1_2.jpg* abgelegt werden usw.
2. Es soll die Datei *bild1.jpg* nach **sortiert/bilder** verschoben werden. Dort existiert noch keine Datei namens *bild1.jpg*, also kann die zu verschiebende Datei ohne vorigem Umbenennen verschoben werden.

Nachfolgend ein Beispiel, wie eine Struktur vor und nach dem Sortieren aussehen soll (die automatisch umbenannten Dateien sind färbig hinterlegt):

vorher

- *index.php*
- *ausgeloggt.php*
- **daten**
 - *bild1.jpg*
 - *bild2.jpg*
 - *bild1.gif*
 - *text1.txt*
 - *text2.docx*
 - **sub1**
 - *bild1.jpg*
 - *bild1.svg*
 - **sub1-1**
 - *bild1.jpg*
 - *text1.txt*
 - **sub1-2**
 - *bild2.jpg*
 - *excel1.xlsx*
 - **sub2**
 - *bild3.gif*
 - *bild4.jpg*
 - *text3.rtf*
 - *text4.docx*
 - **sub2-1**
 - *excel1.xlsx*
 - *bild1.jpg*

nachher

- *index.php*
- *ausgeloggt.php*
- **daten**
 - **sub1**
 - **sub1-1**
 - **sub1-2**
 - **sub2**
 - **sub2-1**
- **sortiert**
 - **bilder**
 - *bild1.jpg*
 - *bild2.jpg*
 - *bild1.gif*
 - *bild1_1.jpg*
 - *bild1_2.jpg*
 - *bild2_2.jpg*
 - *bild3.gif*
 - *bild4.jpg*
 - *bild1_3.jpg*
 - **textdokumente**
 - *text1.txt*
 - *text2.docx*
 - *text1_1.txt*
 - *text3.rtf*
 - *text4.docx*
- **unsortiert**
 - *bild1.svg*
 - *excel1.xlsx*
 - *excel1_1.xlsx*

Nach erfolgreichem Sortieren soll die sortierte Dateistruktur am Bildschirm dargestellt werden.

Der eingeloggte User soll die Möglichkeit bekommen, sich auch wieder ausloggen zu können, wobei er nach erfolgtem Logout auf die Seite *ausgeloggt.php* weitergeleitet werden soll.

2.2 Beispiel 2

Erstellen Sie ein moderiertes Gästebuch mit zugehörigem geschützten Bereich (»Admin-Tool«), wo die Beiträge aktiv oder inaktiv geschaltet bzw. gelöscht werden können.

Ein moderiertes Gästebuch ist ein Gästebuch, wo die Einträge nicht sofort freigeschaltet werden, sondern von einem Administrator freigeschaltet werden müssen – erst nach Freischaltung soll der jeweilige Eintrag im Gästebuch sichtbar werden.

Die Einträge sollen folgende Informationen speichern:

- Name des Eintragsenden *
- Nachricht *
- Datum und Uhrzeit des Eintrags *

Das Administrationstool soll über einen Login erreichbar sein, wobei die Logindaten aus einer Tabelle ausgelesen werden sollen (es soll mehrere Administratoren geben können; Groß- und Kleinschreibung beim Login nur bei dem Passwort wichtig). Im Admin-Tool sollen die Beiträge aufgelistet werden:

- Zuerst werden alle noch nicht freigeschalteten Beiträge aufgelistet (dem Datum nach absteigend sortiert)
- Danach werden alle freigeschalteten Beiträge aufgelistet (ebenfalls dem Datum nach absteigend sortiert)

Die freigeschalteten bzw. nicht freigeschalteten Beiträge sollen entsprechend gekennzeichnet sein (Farben, Gruppierungen, etc.). Der Administrator soll folgende Möglichkeiten haben:

1. Beiträge aktiv und inaktiv schalten
2. Beiträge (dauerhaft) löschen (das Löschen muss bestätigt werden: »Sind Sie sicher, dass Sie diesen Beitrag wirklich löschen wollen?«)

Des Weiteren muss der Admin die Möglichkeit des Logout haben.

2.3 Theoretische Fragen

1. Beschreiben Sie den Unterschied zwischen POST- und GET-Daten.
2. Erklären Sie den Begriff von Sessions und erläutern Sie ein mögliches Einsatzgebiet.
3. Erklären Sie den Begriff einer Datenbank: was ist eine Datenbank, welche grundlegende Unterscheidung trifft man im Falle von Datenbanken (insbesondere

auf die Form der Datenspeicherung). Erklären Sie weiters, was man unter dem Begriff eines »Constraints« versteht, welche Formen von Constraints im Falle von MySQL/MariaDB möglich sind und für welche SQL-Befehle (»Vorgänge«) diese zum Einsatz kommen. Idealerweise geben Sie einige Beispiele an.

4. Erklären Sie die verschiedenen Varianten von JOINS und geben Sie je ein Beispiel an.