

Einführung ins Programmieren

Projekt Mensameter

Kevin Brandao de Graca, 2297954

Fabian Kahlich, 2288758

Inhalt

Aufgabe der Software	3
Installationsanleitung.....	3
Bedienungsanleitung.....	4
Systemarchitektur	5
Technischer Teilaspekt.....	6
Verschlüsselung	6
PHP-Session	6
Fazit.....	7

Aufgabe der Software

Bei unserer Software handelt es sich um ein Bewertungssystem mit vorhergehendem Login und abschließender Durchschnittsberechnung. Für den Login muss sich der Nutzer zuerst mit einem Nutzernamen und einem Passwort auf der Seite registrieren. Diese Daten werden dann verschlüsselt gespeichert, um eine Anmeldung auf der Seite zu ermöglichen. Nach dem Login mit Nutzernamen und Passwort kann der Nutzer zu jedem Produkt aus einem vorher festgelegten Angebot jeweils eine Bewertung zwischen einem und fünf Punkten abgeben, welche dann im System gespeichert und verarbeitet wird. Nach der erfolgreichen Bewertungsabgabe kann sich der Nutzer den Durchschnitt der bisherigen Bewertungen jedes einzelnen Produkts anzeigen lassen, um seine Bewertung für sich selbst einordnen zu können.

Installationsanleitung

Für die Installation des Programmes benötigt man einen Webserver. Wir empfehlen, dafür die XAMPP Anwendung, die eine leichte Installation und Konfigurierung des Apache Webservers ermöglicht. Nach der Installation von XAMPP, kann man nun die Anwendung öffnen und unter den Reiter „Manage Server“ alle Server starten. Zum Speichern von Daten wird eine Datenbank benötigt. Der einfachste Weg diese zu erstellen, ist es nach dem starten der Server in die Adresszeile des Browser localhost/phpmyadmin einzugeben, um auf die phpMyAdmin Seite zu gelangen. Nun kann man sich als erstes unter dem Reiter Benutzerkonten einen neuen Benutzer anlegen. Um möglichst wenig am Code zu ändern nutzen wir „Brandao“ als Benutzernamen, „localhost“ als Hostnamen und „haw“ als Passwort. Bei der Verwendung von anderen Benutzerdaten muss man diese in der Variabel \$verbindung ändern, die sich in allen Dateien außer der logout.php befindet. Bevor man den Benutzer nun erstellt muss man ihn unter dem Punkt „Globale Rechte“ alle Rechte zuweisen und seinen Benutzer durch Drücken des mit „ok“ beschrifteten Buttons erstellen. Nach dem Erstellen eines Benutzers kann man nun damit starten die Datenbank zu erstellen. Links auf der Seite befinden sich die vorhandenen Datenbanken. Klickt man auf den obersten Link „Neu“ kann man eine Datenbank erstellen. Hier gilt nun wieder, um nichts am Code verändern zu müssen, die Nutzung von „htmlworld“ als Datenbankname. Bei Verwendung eines anderen Namens muss man diesen nach der Variable \$verbindung in der Funktion „mysql_select_db“ ändern. Nun hat man eine Datenbank erstellt innerhalb derer man Tabellen erstellen kann. Für unsere Anwendung benötigen wir nur eine Tabelle mit 7 Spalten. Die Tabelle sollten wir „login“ nennen und wie im folgenden Bild ausfüllen.

Name	Typ	Länge/Werte	Standard	Kollation	Attribute	Null	A.I	Kommentare	Virtualität	Spalte verschieben
id	INT	11	Kein(e)		UNSIGNED	<input checked="" type="checkbox"/>				
Aus zentralen Spalten wählen										
user	TEXT		Kein(e)	latin1_swedish_ci		<input type="checkbox"/>				
Aus zentralen Spalten wählen										
password	TEXT		Kein(e)	latin1_swedish_ci		<input type="checkbox"/>				
Aus zentralen Spalten wählen										
productOne	INT	11	Kein(e)			<input type="checkbox"/>				
Aus zentralen Spalten wählen										
productTwo	INT	11	Kein(e)			<input type="checkbox"/>				
Aus zentralen Spalten wählen										
productThree	INT	11	Kein(e)			<input type="checkbox"/>				
Aus zentralen Spalten wählen										
productFour	INT	11	Kein(e)			<input type="checkbox"/>				
Aus zentralen Spalten wählen										

SQL Vorschau Speichern

Nachdem man die Tabelle erstellt hat kann man nun nach dem XAMPP Ordner suchen, in dem sich der htdocs Ordner befindet. In diesen schiebt man nun den Mensameter Ordner hinein und Ruft unter der Adresse localhost/Mensameter/ die Anwendung auf. Die Anwendung ist nun bereit zu Nutzung. Diese Installationsanleitung ist für die Installation auf einem Mac OS Betriebssystem gedacht, aber es sollte mit dieser Anleitung auch möglich sein es auf anderen Betriebssystem zu installieren.

Bedienungsanleitung

Zu allererst muss sich der Nutzer im System registrieren, um einen Abgleich der Daten, die zum Login notwendig sind, zu ermöglichen. Dabei gibt der Nutzer einen Username und ein Passwort an; die Abfrage des Passworts wird zu Sicherheitszwecken wiederholt. Diese Daten werden gespeichert, um später per Abfrage zu ermitteln, ob der Nutzer registriert ist oder nicht. Nach erfolgreicher Registrierung folgt die Anmeldung mit den soeben angegebenen Daten. Nach Vollendung des Logins kann man, wie links zu sehen, die Bewertung für die einzelnen Produkte abgeben und diese durch Drücken des Buttons „Bewertungen abschicken“ an das System übermitteln. Dabei ist zu beachten, dass pro Produkt nur eine Bewertung abgegeben werden kann. Nach der Bewertung der Produkte kann man sich auf die Anzeige der Durchschnittswerte weiterleiten lassen, wo aus sämtlichen bis zum Zeitpunkt des Aufrufs abgegebenen Bewertungen der Durchschnitt berechnet wird. Dieser wird für jedes Produkt einzeln berechnet und ausgegeben. Nach der Betrachtung des Durchschnitts kann man sich wieder ausloggen, indem

Username: kevin
 Passwort:
 Passwort wiederholen:
 Registrieren

Bitte logge dich ein:
 Username: kevin
 Passwort:
 Einloggen

[Jetzt registrieren...](#)

Willkommen kevin

[Logout](#)

Bewertung:

Produkt 1

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

Produkt 2

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

Produkt 3

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

Produkt 4

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

[Bewertungen abschicken](#)

Produkt 1
Durchschnitt: 2.25

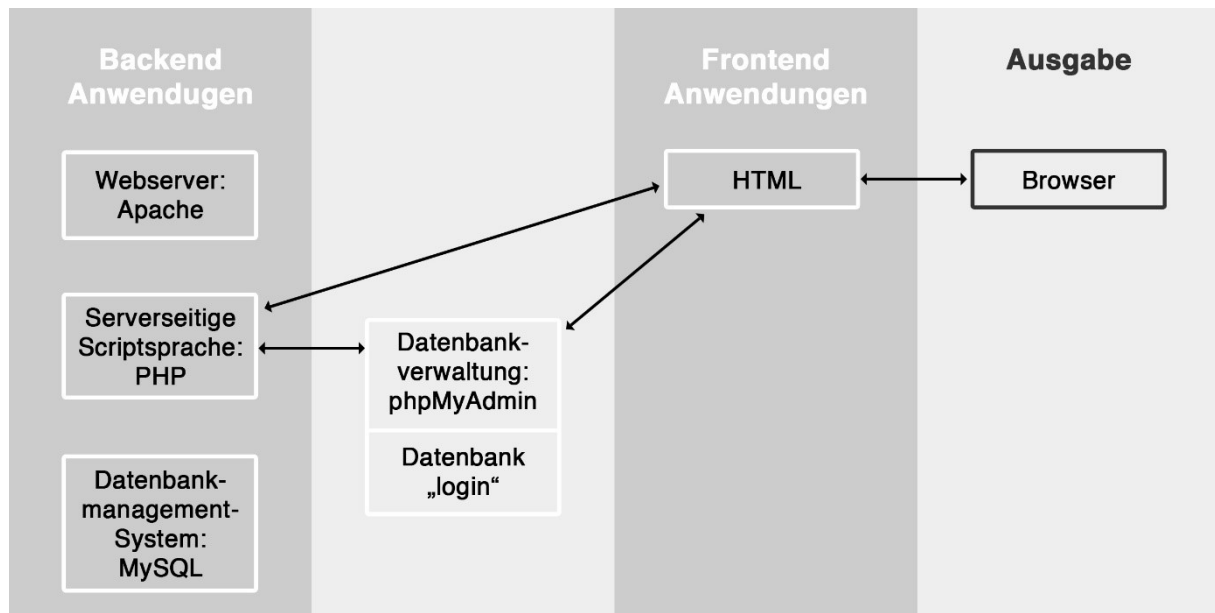
Produkt 2
Durchschnitt: 2.75

Produkt 3
Durchschnitt: 3.75

Produkt 4
Durchschnitt: 2.25

man den mit „Logout“ bezeichneten Button drückt. Daraufhin wird man zu einer Logout-Seite weitergeleitet, um dem Nutzer anzuzeigen, dass dieser wirklich ausgeloggt ist.

Systemarchitektur



Bei der Systemarchitektur unterscheiden wir zwischen Backend Anwendungen und Frontend Anwendungen. Bei unserem System ist typischerweise das Backend näher am System und das Frontend näher am Benutzer.

Das Backend startet mit dem Webserver von Apache, den wir mit Hilfe von XAMPP installiert und konfiguriert haben. XAMPP bietet uns nützliche Werkzeuge, wie zum Beispiel phpMyAdmin. Mit diesem Datenbankverwaltungs-Werkzeug können wir Datenbanken erstellen, verwalten und auslesen. Der Apache Webserver ist einer der meist genutzten Webserver im Internet. Der Webserver bietet die Möglichkeit mit Skriptsprachen wie PHP eine dynamische Webseite zu erstellen. Eine besondere Eigenschaft am Apache Webserver ist, dass man mit Hilfe von verschiedenen Modulen die verschiedenen Stärken der Betriebssysteme ausnutzen kann, um eine erhöhte Stabilität und höhere Geschwindigkeit des Servers zu gewährleisten. Bei unserem System haben wir uns für die Scriptsprache PHP entschieden, um eine dynamische Anwendung zu verwirklichen, die aktuelle Werte darstellen kann. Der Webserver arbeitet nun mit PHP und kann unseren geschriebenen Code verarbeiten. Mit Hilfe von PHP können wir zum Beispiel für den Durchschnitt Werte von der Datenbank bekommen oder können Logindaten in der Datenbank speichern. Beim Auslesen von den Datenbankdaten bekommt PHP diese Daten erst von der Datenbank und gibt diese dann an das Frontend weiter. Beim Speichern von Daten läuft dieser Prozess andersherum ab. Der Nutzer gibt die Daten im Frontend ein, die Daten kommen vom Frontend mit Hilfe von PHP in die Datenbank. Im PHP muss man dafür allerdings die genauen Funktionen und Orte definieren. Im Frontend wird aus den PHP Dateien ein

Quelltext erstellt. Der Quelltext besteht aus HTML Tags, die je nach Inhalt variieren. Um den HTML Code nun Anzeigen zu können wird in der Regel ein Browser genutzt. Bei unserer Anwendung ist es nicht so wichtig für welchen man sich entscheidet. Für die visuelle Darstellung ist selbstverständlich ein Monitor die Voraussetzung.

Technischer Teilaspekt

Verschlüsselung

Wir nutzen den „Message-Digest Algorithm 5“ (kurz MD5), um die vom Nutzer angegebenen Daten nicht als einfachen innerhalb der Datenbank zu speichern. Der MD5 wandelt dabei eine beliebige Nachricht in einen 128-Bit-Hashwert um. Diese Nachricht wird dann als 32-stellige Dezimalzahl gespeichert. Als Beispiel nehmen wir einmal das Wort „Suppe“ und das Wort „Sippe“, die sich in einem

Suppe	Sippe
d438262b393ebdc93b912fc9f610ac5	22d36158a325667a2a07e1a3cfe8db5f

Vokal unterscheiden. Wie man der Tabelle entnehmen kann, unterscheiden sich die Verschlüsselungen der beiden Worte sehr voneinander, während sich die eigentlichen Wörter sehr ähneln. Der MD5 wird heutzutage nicht mehr als sicher anerkannt, was womöglich daran liegt, dass dieser im Jahre 1991 entwickelt wurde. Jeder moderne Rechner findet innerhalb weniger Sekunden eine sogenannte Kollision, also eine beliebige Zeichenfolge, die jedoch die selbe Hexadezimalzahl wie die zu knackende Passage aufweist. Damit kann man zwar nicht auf die ursprüngliche Zeichenfolge zurückschließen, allerdings gelangte es einem Team von Hackern im Jahre 2008 durch einen Kollisionsangriff ein gefälschtes und vertrauenswürdiges CA-Zertifikat zu erstellen, womit sie theoretisch jedes SSL-Zertifikat fälschen hätten können und somit die Sicherheitsmaßnahmen von HTTPS auszuschalten. Dies war jedoch die am einfachsten zu implementierende Verschlüsselung, sodass wir uns für diese entschieden, um die Daten nicht als Fließtext zu speichern.

PHP-Session

Mit dem Starten unserer Anwendung wird als aller erstes ein Cookie auf dem Rechner abgelegt. Den Cookie erstellen wir mit dem Befehl `session_start()`. Der Cookie ist ein zufällig erstellter Code, mit der ID "PHPSESSID". Mit diesem Cookie kann man zum Beispiel überprüfen ob der Nutzer eingeloggt ist oder nicht. Hinter diesem Cookie, wird der Nutzernamen abgespeichert. Anhand des Nutzernamens kann man auch erkennen, ob die Person eingeloggt ist oder nicht. Wenn der Nutzernamen vorhanden ist, ist die Person eingeloggt ansonsten nicht. Nachdem der Nutzer sich eingeloggt hat, gelangt er in den geschützten Bereich. Der geschützte Bereich muss mit dem Befehl `session_start()` starten, um den Cookie fortlaufend zu speichern. Ohne diesen Befehl wäre es möglich, die Seite auch ohne Login zu

erreichen. Im weiteren Verlauf kann man mithilfe der Session den Nutzernamen aus dem Cookie wieder ausgeben.

Fazit

Dadurch dass wir in der Vorlesung nur sehr sporadisches Wissen bezüglich HTML, PHP, MySQL und deren Verknüpfung erhalten haben, mussten wir uns das meiste selbst aneignen. Dies führte dazu, dass wir erst einmal viel Zeit für das theoretische Verständnis dieser Programmiersprachen aufwenden mussten, was den Zeitaufwand natürlich um einiges in die Höhe trieb. Nachdem wir uns in die Materie eingearbeitet hatten, mussten wir wiederum einen Plan erstellen, wie wir die Anwendung konzipieren möchten. Diese beiden notwendigen Schritte führten dazu, dass wir circa ein Drittel unserer gesamten Zeit für die Vorbereitung der Anwendung aufwenden mussten. Das eigentliche Programmieren gestaltete sich dann relativ zügig, auch wenn wir immer wieder mit kleinen Problemen zu kämpfen hatten, die wir aber immer recht schnell durch gemeinsames Brainstorming beheben konnten. Hin und wieder mussten wir uns zwar Inspiration aus dem Internet einholen, jedoch betraf dies meist nur PHP-Operatoren, die wir noch nicht genutzt haben und deswegen kein Wissen darüber hatten. So hatten wir lange Zeit das Problem, dass die abgestimmten Werte innerhalb der Datenbank nicht in die Zeile des zugehörigen Nutzers eingetragen wurden, sondern in eine eigenständige Zeile eingetragen wurden. Dies lag daran, dass wir anstatt des Operators „*UPDATE*“ den Operator „*INSERT INTO*“ nutzten. Ein wiederkehrendes Problem konnten wir jedoch nicht beheben, nämlich dass jeder Nutzer nur einmal abstimmen kann. Zwar werden bei erneuter Werteeingabe die alten Werte ersetzt, jedoch wollten wir eigentlich nur eine einmalige Stimmenabgabe ermöglichen. Wir dachten daran, über eine if-Schleife eine Überprüfung der eingespeicherten Werte einzufügen, welche im Falle bereits bestehender Werte eine Meldung ausgibt, dass man bereits einmal abgestimmt hat, allerdings sind wir daran leider gescheitert, da man die Datenbank nicht nach dem Wert 0 absキャン kann, da dieser nicht vorkommt, da ein standardmäßiger Wert 0 für jeden Wert die Durchschnittsberechnung verfälschen würde. Ein ähnliches Problem haben wir bei nicht erfolgter Stimmenabgabe festgestellt. Sobald eine weitere Nutzerzeile angelegt wird, wird die Durchschnittsrechnung automatisch angepasst, was zur Verfälschung der Durchschnittswerte führt. Wir hatten uns überlegt, dies über eine Abfrage während der Stimmabgabe zu verhindern, indem wir eine Stimmabgabe verhindern, sollten noch nicht alle Werte eingetragen sein. Andernfalls könnte man das Problem lösen, indem man eine Überprüfung der einzelnen Zeilen einrichtet, die jede Zeile, die nicht vollständig ausgefüllt ist, aus der Berechnung des Durchschnitts ausschließt, jedoch wissen wir nicht wie wir diese Operation einfügen sollen.