Dokumentation

Studentenveranstaltungsforum „StuFo“

06.02.2018

# Teammitglieder und ihre Rollen

|  |  |
| --- | --- |
| Kevin Aulbach: | Entwickler im Bereich PHP, SQL und Datenbank |
| Lukas Becker: | Teamleiter und Entwickler im Bereich HTML |
| Julian Henke: | Entwickler im Bereich HTML |
| Louisa Viktoria Schmitt: | Design mit CSS, sowie die Gestaltung des Logos |

# Projektbeschreibung

Motivation

Durch die Überlegungen wie genau wir Studenten semester- und studiengangsübergreifend vernetzten können, sind wir auf die Idee eines Veranstaltungsforums für Studenten gekommen. In diesem Studentenveranstaltungsforum können Studenten Veranstaltungen erstellen, sodass andere angemeldete Studenten daran teilnehmen können. Die Veranstaltungen umfassen Freizeitaktivitäten, den Bereich Sport und Kultur, sowie das Treffen zum Lernen.

Zielsetzung

Das Ziel des Projekts „Studentenveranstaltungsforum“, kurz „StuFo“, ist die Vernetzung von Studenten der Hochschule Hamm-Lippstadt per Veranstaltungen. Der Anwendungsbereich, welcher auf Studenten zugeschnitten ist, soll diesen ermöglichen Veranstaltungen zu erstellen, zu bearbeiten, beizutreten und zu löschen. Das Forum verfolgt die Ziele Studenten über Veranstaltungen miteinander zu verbinden, sodass sie in persönlichen Kontakt treten können. Weitere Informationen können die Nutzer im Bereich unter der Veranstaltungsbeschreibung austauschen. Die Webseite soll überwiegend auf PCs laufen, kann aber auch aufs Smartphones abgerufen werden. Unsere Zielgruppe sind Studenten und Studentinnen der Hochschule-Hamm-Lippstadt. Die Website erfordert keine spezifischen Hardwareeigenschaften, bis auf einen Internetzugang.

Lösungsansatz

Hauptsächlich besteht das Projekt aus mehreren PHP-Seiten unter Verwendung von HTML 5, die miteinander verknüpft sind. Das Design ist mit CSS Version 4 implementiert beziehungsweise designt worden. Das Backend stellt eine MySQL-Datenbank, in die Nutzerdaten, Veranstaltungen, Beiträge, sowie Teilnahmen an den Veranstaltungen eingetragen werden und PHP-Funktionen, die die Input-Felder der Formulare auslesen und diese in SQL-Befehle übergeben. Um die Veranstaltung, die Beiträge und die Teilnehmerliste zu sortieren werden ebenfalls SQL-Befehle verwendet, um diese dann sortiert nachdem jeweiligen Parameter auszugeben.

# State of the Art und Related Work

Heutzutage gibt es allerhand Foren zu jedem Bereich, der dem Menschen von Interesse ist. Ein Beispiel für die Wichtigkeit von Veranstaltungen zeigt Facebook mit ihrem Bereich für Veranstaltungen. Der Unterschied zu diesem Projekt ist, dass bei „StuFo“ die Teilnehmerzahl festgelegt werden kann und sichergestellt wird, dass nur Studenten der Hochschule-Hamm-Lippstadt teilnehmen können.

# Projektplan Soll / Ist

Das Projekt wurde in 6 Phasen unterteilt: dem Kick-Off, der Konzeption, der Vorbereitung, der Entwicklungsphase, die Phase des Testens und der Inbetriebnahme, so wie schließlich die Dokumentation und Abgabe. Der zeitliche Aufwand des Projekts entspricht den in der Excel-Tabelle angegebenen 19 Wochen. Der Aufwand wurde unter den Teammitglieder circa gleich aufgeteilt, sodass jedes Mitglied seinen Teil beitragen konnte.



# Anforderungsanalyse

Das Studentenveranstaltungsforum hat die Anforderung, dass Nutzer sich mit einer E-Mail-Adresse der Hochschule Hamm-Lippstadt anmelden. Die Überprüfung findet durch die Endung „@stud.hshl.de“ statt. Die Registrierung beziehungsweise der Log-in erfolgt über die Eingabe der „@stud.hshl.de“ E-Mail-Adresse und einem Passwort, welches einen Großbuchstaben und 8 bis 20 Zeichen enthalten muss. Das Passwort kann nach der Anmeldung geändert werden. Jeder Nutzer kann sich nur einmal mit seiner E-Mail-Adresse registrieren. Jedes Konto enthält einen Vornamen, Nachname, das Geschlecht, das Alter, den Studiengang und ein optionales Profilbild. Der Nutzer kann sein Profilbild auf der Profilseite hochladen und ändern. Die Kontodaten werden in eine Datenbank eingetragen und von dort abgefragt. Die Änderung des Passworts erfolgt über Eingabe des alten beziehungsweise neuen Passworts. Das Studentenforum bietet die Funktion Veranstaltungen zu erstellen, beizutreten und auszutreten. Nach Erstellung einer Veranstaltung durch den Nutzer kann nur dieser die Veranstaltung unter doppelter Eingabe des Passworts löschen. Die Veranstaltungen enthalten den Namen, das Datum, die Uhrzeit, den Ort, den Ersteller, eine Kategorie und eine Beschreibung, die auf 280 Zeichen begrenzt ist. Der Ersteller kann eine maximale Teilnehmerzahl angeben. Die resultierende Teilnehmerliste hat eine Filterfunktion nach den Kriterien: zuletzt beigetreten, Nachnamen, Vornamen und Geschlecht. Die Startseite verfügt über eine Navigationsleiste auf der linken Seite zum schnellen Navigieren durch die Kategorien. Nach dem Beitreten zu einer Veranstaltung, kann der Nutzer einen Beitrag verfassen, der auf 280 Zeichen begrenzt ist. Dadurch können die Nutzer miteinander kommunizieren. Ein Beitrag enthält nach dem Abschicken den Namen des Verfassers, einen Zeitstempel und das Profilbild (optional). Die Beiträge befinden sich nach dem neusten Zeitstempel sortiert unter der Veranstaltungsbeschreibung. Die fünf neusten Beiträge werden direkt angezeigt. Möchte der Nutzer ältere Beiträge lesen, muss dieser auf den Button „mehr…“ klicken. Das Forum soll möglichst 24 Stunden am Tag online sein. Die Webseite ist auf PCs gut lesbar dargestellt. Ebenso sind Reaktionszeiten bei hoher Belastung gering. Das Forum ist nutzerfreundlich gestaltet, sodass Studenten und Studentinnen sich direkt zurechtfinden.

# Konzeption

Durch unser Studentenveranstaltungsforum („StuFo“) wollen wir neue Freundschaften, Bekanntschaften, Lerngruppen und Sportbegeisterte zusammenführen. Wir wollen damit erreichen, dass sich Studenten aus allen Studiengängen und nicht nur aus demselben Studiengang näherkommen.

# Entwurf und Umsetzung

Anwendungsfalldiagramm:

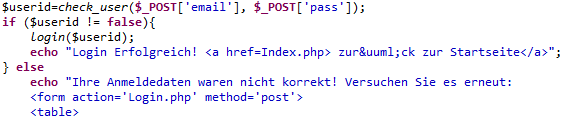
**

**

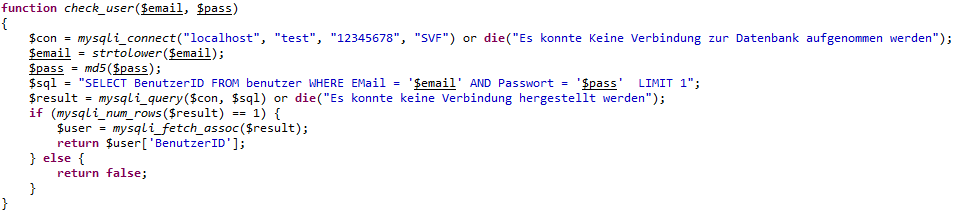
Beschrieben wird die Funktionalität der Kontoverwaltung. Der Nutzer hat 3 Möglichkeiten sein Konto zu bearbeiten. Er kann sein komplettes Konto löschen, sein Passwort ändern und sein Profilbild ändern. Wenn ein Nutzer sein Konto löschen möchte, muss dieser sein Passwort eingeben. Anschließend wird das eigegebene Passwort mit dem in der Datenbank befindlichem Passwort abgeglichen. Ist das eingegeben Passwort ungültig, so wird der Nutzer aufgefordert das Passwort erneut einzugeben. Ist es andernfalls gültig, so wird das Konto gelöscht. Wenn der Nutzer sein Passwort ändern möchte, wird er gebeten sein altes Passwort einzugeben. Ist dieses ungültig, heißt nicht übereinstimmend zu dem in der Datenbank, wiederholt sich der Prozess. Falls das alte Passwort korrekt ist, wird der Nutzer aufgefordert ein neues Passwort mit einem Großbuchstaben und 8 bis 20 Zeichen einzugeben. Wird dieses nicht korrekt durchgeführt, muss der Nutzer das neue Passwort erneut eingeben. Andernfalls ist das neue Passwort gültig und es wird in der Datenbank überschrieben. Falls der Nutzer sein Profilbild ändern möchte, wählt dieser „Profilbild ändern“ bzw. hochladen. Nach dem Auswählen eines Bildes wird geprüft, ob das Format passt und die maximale Bildgröße von 250kb nicht überschritten wird. Ist die Eingabe ungültig wird der Nutzer aufgefordert ein anderes Bild hochzuladen, andernfalls wird das Bild auf dem Server und der Pfad in der Datenbank abgespeichert.

Für die Anwendung wird Xampp benötigt, um einen lokalen Webserver zu benutzen (siehe XAMPP Anleitung).

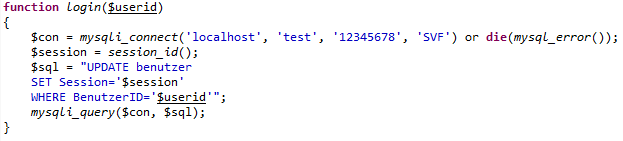
Eine bedeutende Funktion der Webseite ist die Möglichkeit sich anzumelden, dabei können wir den genauen Quellcode betrachten:



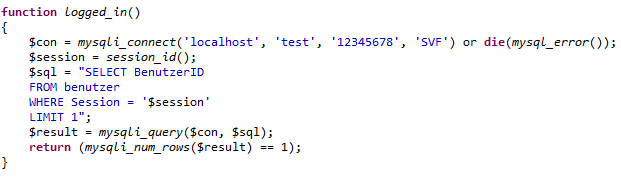
Wir betrachten zunächst die Hauptmethode des Logins, dort wird zunächst die Funktion „check\_user“ ausgeführt. Hier wird die im Input-Feld stehende E-Mail und das Passwort an die Funktion übergeben. Dabei wird die “BenutzerID” des Benutzers, falls dieser existiert zurückgegeben und in der Variable „$userid“ gespeichert. Falls diese nicht „false“ zurückgibt, wird die Funktion „login“ mit der „BenutzerID“ ausgeführt, welche die aktuelle „SessionID“ des Benutzers in die Datenbank speichert.



Der Funktion “check\_user” wird zunächst die E-Mail und das Passwort des einloggenden Benutzers übergeben. Dabei wird in der Variable „$con“ der Login der Datenbank übergeben. Danach wird die E-Mail mit der Funktion „strtolower“ in Kleinbuchstaben abgespeichert und das Passwort über die Funktion „md5“ verschlüsselt, da das Passwort in der Datenbank verschlüsselt eingespeichert ist. Als nächstes wird der SQL-Befehl, welcher die „BenutzerID“ anzeigt, die die angegebene E-Mail und das Passwort enthält, in der Variablen gespeichert. Nun wird in der Variable „$result“ mit der Funktion „mysli\_query“ der SQL-Befehl und die Connection zur Datenbank übergeben, sodass der SQL-Befehl, sobald die Variable aufgerufen wird auf der angegebenen Datenbank ausgeführt wird. Danach wird durch eine If-Funktion überprüft, ob genau eine Zeile als Ergebnis herauskommt (Da es nur eine E-Mail mit dem passenden Passwort gibt). Ist dies der Fall übergibt die Funktion „mysqli\_fetch\_assoc“ die Einträge der Datenbank in die Variable „$user“ und zum Ende wird dabei die „BenutzerID“ des Benutzers zurückgegeben. Werden bei der If-Anweisung mehr oder weniger Zeilen als eine erkannt, gibt die Funktion „false“ zurück.



Der Funktion „login“ wird die Variable „$userid“ übergeben. Als erstes wird die Verbindung zur Datenbank unter der Variablen „$con“ gespeichert. Danach wird die aktuelle „SessionID“ des Benutzers in der Variable „$session“ gespeichert. Des Weiteren wird der SQL-Befehl, bei der Tabelle Benutzer mit der übergebenen „BenutzerID“ die aktuelle „SessionID“ eingetragen wird. Dieser SQL-Befehl wird dann nur noch über die Funktion „mysqli\_query“ mit der Verbindung und dem SQL-Befehl ausgeführt. Als nächstes betrachten wir, wie der Inhalt an bestimmten Stellen nur für eingeloggte Mitglieder verfügbar ist. Dies wird durch die Funktion „logged\_in“ ermöglicht:



Bei der Funktion „logged\_in“ wird zunächst wieder die Verbindung zur Datenbank, sowie die aktuelle „session\_id“ in einer Variable abgespeichert. Danach wird bei dem SQL-Befehl die „BenutzerID“ gesucht, der der aktuellen „SessionID“ entspricht und dabei auf 1 limitiert wird, da es sowieso nur eine gleiche „SessionID“ in der Datenbank geben sollte. Danach wird wieder das Ausführen des Befehls in der Variable „$result“ gespeichert und als Rückgabewert wird „true“ für die Abfrage geliefert, falls diese genau eine Zeile zurückgibt.

# Evaluation

Das Durchführen von Tests wurde durch Eingabe verschieden großer Texte in den Eingabebereichen realisiert. Dabei wurde darauf geachtet, ob die Tabellenbreite die maximale Seitenbreite nicht überschreitet. Außerdem wurde geprüft, ob alle Funktionalitäten richtig funktionieren. Alle Tests waren erfolgreich.

# Zusammenfassung und Ausblick

Das Projekt erfüllt einen großen Teil der im Pflichtenheft aufgeführten Anforderungen. Zukünftig können weitere Funktionen hinzugefügt werden und Überarbeitungen im Bereich Design durchgeführt werden, um die Webseite noch benutzerfreundlicher und verständlicher zu machen. Des Weiteren könnte die Weiterleitung auf die aktuelle Seite verbessert werden, sodass der Nutzer eine bessere User Experience hat.

# Literatur

<http://wiki.selfhtml.org/wiki/HTML>

<https://www.w3schools.com/html/>

<https://www.gimp.org/>