Inhalt

[Literaturverzeichnis II](#_Toc515611825)

[Abbildungsverzeichnis III](#_Toc515611826)

[1. Allgemeines - 1 -](#_Toc515611827)

[1.1 Aufgabenstellung - 1 -](#_Toc515611828)

[1.2 PHP - 1 -](#_Toc515611829)

[1.3 Structured Query Language - 4 -](#_Toc515611830)

[2. Eigene Ansätze / Vorbereitungen - 4 -](#_Toc515611831)

[2.1. Weitere Vorüberlegungen - 5 -](#_Toc515611832)

[3. Überlegungen während des Projektes - 6 -](#_Toc515611833)

[4. Test durch unabhängige 3. Person - 7 -](#_Toc515611834)

[5. Fazit - 8 -](#_Toc515611835)

# Literaturverzeichnis

1. <http://php.net/>
2. https://de.wikipedia.org/wiki/PHP
3. <https://de.wikipedia.org/wiki/SQL>

# Abbildungsverzeichnis

[Abbildung 1: PHP-Logo - 1 -](#_Toc515611836)

[Abbildung 2: Architektur - 3 -](#_Toc515611837)

[Abbildung 3: Mind-Map - 4 -](#_Toc515611838)

[Abbildung 4: Eine farbliche Meldung ersetzt das Popupfenster und ermöglicht ein unterbrechungsfreies Laden der Seite - 7 -](#_Toc515611839)

[Abbildung 5: Vor der Änderung erschien nur diese Meldung bevor wieder das leere Formular angezeigt wurde - 8 -](#_Toc515611840)

[Abbildung 6: Nach den Änderungen - 8 -](#_Toc515611841)

# 1. Allgemeines

## 1.1 Aufgabenstellung

Im Rahmen des Studienmoduls Managementtechniken im IT-Projekt (MTITP) gilt es eine vorgegebene Aufgabe bzw. Problemstellung mit einer bestimmten Technologie im Bereich der Informatik, als Gruppenaufgabe, zu lösen. Diese Aufgabe erstreckte sich über 2 Theoriesemester und ein Praxissemester.

Es wurde von den Studenten gefordert, eine Webanwendung für die Beantwortung von Schätzfragen zu erstellen. Diese Aufgabe sollte möglichst mit der Technik „Personal Home Page Tools“ oder kurz „PHP“ gelöst werden.

## 1.2 PHP

PHP ist eine Scriptsprache, deren Syntax an C und Perl angelehnt ist. Es wird vorwiegend zur Erstellung dynamischer Webseiten und Webanwendungen verwendet. Die Sprache bietet eine breite Datenbankunterstützung sowie Internet-Protokolleinbindung und zahlreiche Funktionsbibliotheken. Seit der Veröffentlichung 1995 wurden umfangreiche Änderungen in der Sprache vorgenommen. Die aktuellste Version ist PHP7.2, welche seit dem 30. November 2017 öffentlich ist. Diese Version zeichnet sich vor Allem durch eine bis zu 30% schnellere Ausführungszeit gegenüber dem Vorgänger PHP5. Hierzu ist zu sagen, dass keine Version mit dem Namen PHP6 veröffentlich wurde, da diese als gescheitert gilt. Das Entwicklerteam von PHP entschied sich bei der Namensgebung für den Nachfolger von PHP5 daher zum Überspringen einer Versionsnummer, was zum mittlerweile bekannten PHP7 führte. Im Folgenden werden die wichtigsten Versionen von PHP vorgestellt und die wichtigsten Neuerungen genannt.



Abbildung 1: PHP-Logo

**PHP1 (Personal Home Page Tools)**

Wie bereits erwähnt wurde diese erste Version im Jahre 1995 veröffentlicht. Zunächst war PHP als eine Sammlung von Perl-Skripten gedacht, welche vom Entwickler Lerdorf zur Protokollierung von Zugriffen auf eine Onlineanwendung geschaffen wurde.

**PHP2**

Diese Version ähnelt der ersten Version sehr stark, unterscheidet sich allerdings in der Tatsache, dass hierbei die Programmiersprache C als Basis genutzt wird. Eine Bedeutung dieser Version liegt darin, dass sie die Grundlage der weiteren Versionen darstellt und bis heute auf dieser Basis weiterentwickelt wird. Sie ist somit also als Grundlage der Heute im Einsatz befindlichen Versionen zu sehen.

**PHP3 / PHP4**

Unter diesen beiden Versionen wurde die Sprache weiterentwickelt und erreichte einen höheren Bekanntheitsgrad. In PHP3 wurden die Inhalte der ersten beiden Versionen komplett neu geschrieben um eine bessere Nutzung für den E-Commerce-Bereich zu ermöglichen. Des Weiteren wurden unter der 4 Version neben der Nutzung der Zend Engine 1 (Entwicklung der Zend Technologies Ltd.) als Kern der Sprache einige Verbesserungen vorgenommen. Hierzu zählen besonders die Verbesserung der Ausführungsgeschwindigkeit, das Sessionmanagement, eine Ausgabenpufferung und die Unterstützung für viele weitere Webserver.

**PHP5**

Eine der bekanntesten und bis heute wahrscheinlich am verbreitetsten Versionen ist PHP5, welches im Sommer 2004 veröffentlicht wurde. Da bis zur Veröffentlichung der Nachfolgerversion über 11 Jahre vergangen sind, ist diese Version sehr verbreitet und wird in unzähligen Webanwendungen verwendet.

Vorteil gegenüber dem Vorgänger ist die Nutzung der Zend Engine II, welche ein besseres Objektmodell vorweisen kann. Dies ermöglicht eine effizientere Entwicklung von objektorientierten Anwendungen und der damit Verbundenen Komponenten und Sprach Konstrukte. Durch PHP5 wurde ein objektorientiertes Programmieren in PHP erstmals interessant für die meisten Entwickler, obwohl auch bereits Vorgängerversionen eine objektorientierte Entwicklung ermöglicht hätten. Genauere Informationen zu den Neuerungen der jeweiligen Versionen (PHP5.1, 5.2, …) würden den Rahmen dieser Betrachtung übersteigen, man kann allerdings zusammenfassend festhalten, dass die Unterstützung des objektorientierten Ansatzes stetig vorangetrieben wurde und eine Arbeit mit Objekten in PHP für die Entwickler immer einfacher und lukrativer gestaltet wurde.

**PHP7**

Aufgrund des Scheiterns der sechsten Entwicklungsstufe wurde wie bereits erwähnt die Entscheidung getroffen, die neue Version mit dem Namen PHP7 zu versehen.

Da unter anderem Hashtabellen neu implementiert wurden, konnte im Vergleich zum Vorgänger PHP5 eine bis zu 30% geringere Ausführungszeit erzielt werden. Weitere Optimierungen brachten zudem einen geringeren Speicherbedarf und neue Features, wie beispielsweise Engine-Exceptiones oder anonyme Klassen.

**Funktionsweise**

Bei PHP wird der Code serverseitig verarbeitet. Der Quelltext wird daher nicht an den Browser übermittelt, sondern an den Webserver, wo er von einem Interpreter verarbeitet wird. Lediglich die Ausgabe (sofern vorhanden) wird an den Webbrowser übermittelt und von diesem dem für den Nutzer dargestellt. Üblicherweise erfolgt dies über ein HTML-Dokument, wobei allerdings auch die Generierung anderer Dateitypen wie beispielsweise PDF von PHP unterstützt wird. Ein PHP-Skript lässt sich direkt in HTML integrieren, was theoretisch das Abspeichern als PHP-Datei überflüssig macht.



Abbildung 2: Architektur

Jede PHP-Seite belastet den Webserver, wobei der Quelltext bei jedem Aufruf der Seite erneut verarbeitet wird. Ausnahme hierbei bilden dynamische Frontends, bei denen lediglich ein Container beim Laden der Seite erzeugt wird. Dieser kann dann beliebig mit Inhalten gefüllt werden, ohne dass die Seite und somit auch der Quellcode neu geladen und verarbeitet werden muss. Näheres zur Funktionsweise dieser Technologie werden an dieser Stelle allerdings nicht betrachtet, da bei dem vorliegenden Projekt eine so komplexe Mechanik nicht zum Einsatz gekommen ist.

Um die Serverlast zu reduzieren, welche beim ausführen multipler PHP-Anfragen entsteht, können diverse Cache-Systeme genutzt werden, welche beispielsweise Abfrageergebnisse von Datenbankabfragen vorhalten und somit ein erneutes Ausführen unnötig machen. Außerdem ist es möglich eine zum ausführen bereite Version des Programmes im Cache zu halten, was ein neu Laden des Quellcodes überflüssig werden lässt.

Obwohl PHP auch das Schreiben von komandozeilenorientierten Skripten ermöglicht und sogar eine Programmschnittstelle für grafische Oberflächen zur Verfügung steht, wird PHP weiterhin vorrangig für Webanwendungen genutzt.

## 1.3 Structured Query Language

Structured Query Language oder auch kurz SQL (zu Deutsch: strukturierte Abfragen-Sprache) ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken. SQL ermöglicht zudem das Bearbeiten (Einfügen, Verändern und Löschen) sowie die Abfrage von Daten aus dem Datenbestand.

Die Basis von SQL ist die relationale Algebra, wobei die Sprache selbst an die englische Umgangssprache angelehnt ist. Die Sprache wurde unter Mitwirken einiger Normierungsgremien wie ANSI oder DIN standardisiert. Der Einsatz von SQL bietet einem den Vorteil der Unabhängigkeit der Anwendung vom eingesetzten Datenbankmanagementsystem, weshalb die Sprache in fast allen gängigen Datenbanksystemen zum Einsatz kommt (jeweils mit kleineren Abwandlungen).

SQL-Befehle werden in drei Kategorien unterteilt, die im Folgenden aufgeführt werden.

DML - Befehle zur Datenmanipulation (z.B. Anlegen, Löschen, Ändern und Abrufen von Daten)

DDL - Befehle zur Definition des Datenbankschemas (z.B. Aufbau von Tabellen und Definition von Relationen)

DCL - Befehle für Rechte- und Transaktionskontrolle (z.B. Definition von Zugriffsrechten auf Datenbestände durch bestimmte Datenbanknutzer)

Mit SQL kann wie bereits erwähnt eine Vielzahl von Zugriffen auf die Datenbasis ermöglicht werden. Dazu werden die sogenannten Abfragen benutzt, die im Prinzip einen recht simplen, strukturell logischen Aufbau aufweisen. Dem Nutzer ist es allerdings auch möglich komplexere SQL-Skripte zu entwerfen, die dann eine Vielzahl von Abfragen und Funktionen zur Verarbeitung der Daten beinhalten können. Hierbei bietet sich dem Nutzer der Vorteil, dass das ausführen eines entsprechend durchdachten SQL-Skriptes eine wesentlich bessere Performance und wesentlich mehr Möglichkeiten zur Abfrage und Aufbereitung der Daten bietet, als wenn die gleiche Anforderung mit beispielsweise einem PHP-Skript durchgeführt werden würde. Das folgende Beispiel zeigt eine einfache Abfrage, wie sie häufig zum Abfragen von Daten aus einer Bestimmten Tabelle genutzt wird.

SELECT \*

FROM Student;

Im Falle dieses Beispiels würden dem Nutzer alle Einträge der Tabelle „Student“ zurückgeliefert werden. Durch genaue Angabe von bestimmten Spalten hinter der SELECT-Anweisung statt des \* könnte das Ergebnis noch auf bestimmte Spalten der Tabelle eingegrenzt werden.

# 2. Eigene Ansätze / Vorbereitungen

In diesem Kapitel werden die ersten Vorbereitungen, welche getroffen wurden, erklärt um die Aufgabenstellung aus 1.1 zu bewältigen.

Zu allererst hat sich die Gruppe, bestehend aus 3 Studenten, zusammengesetzt und eine Mind-Map mit Ihren ersten Ideen und Eindrücken erstellt.

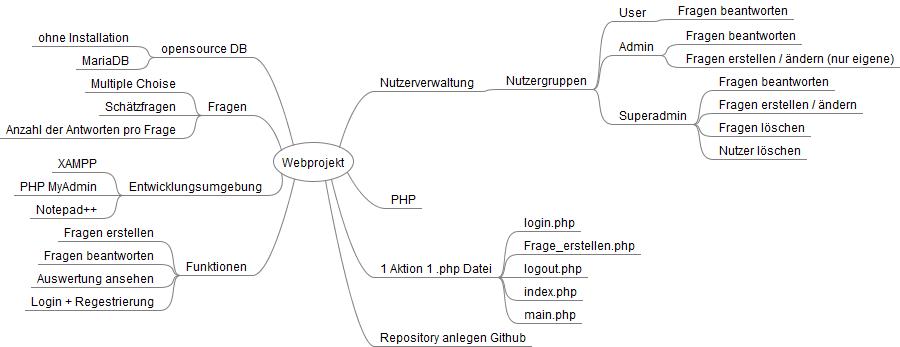


Abbildung 3: Mind-Map

Um diese Aufgabe als Gruppe ordentlich lösen zu können brauchte man ein Repository, welches über die Website „Github.com“ lief. Ein Repository wird immer dann benötigt, wenn mehrere Leute gleichzeitig an einem Projekt arbeiten, um die Organisation und den Ablauf der Programmierung erleichtern zu können.

Als nächstes machte man sich Gedanken, was die Webanwendung später einmal können soll, bzw. was für Funktionen diese haben muss. Als Gruppe kam man auf folgende Funktionen die diese Webanwendung auf jeden Fall können muss. Man sollte sich einloggen und Registrieren können, Fragen sollten nach freien belieben erstellt und beantwortet werden können und die Webanwendung sollte in der Lage sein eine Auswertung über die beantworteten Fragen geben zu können.

Weiterhin braucht es eine Nutzerverwaltung mit Nutzergruppen. Nutzergruppen wären in diesem Fall der allgemeine User, der Admin und der Superadmin. Der allgemeine User kann lediglich Fragen beantworten und eine Auswertung über seine eigenen gegebenen Antworten sehen. Der Admin ist als Nutzergruppe dafür zuständig Fragen erstellen bzw. seine eigenen Fragen auch ändern und beenden zu können, zusätzlich kann er ebenfalls Fragen beantworten. Als Letztes der Superadmin, diese Nutzergruppe darf zusätzlich zu den Rechten des Admins noch Fragen löschen und Nutzer löschen.

Der nächste Punkt auf der Mind-Map war welche Arten von Fragen könnten alles aufkommen, dazu hat man sich auf Multiple Choice Fragen und Schätzfragen erstmal festgelegt.

Der nächste Gedanke bezog sich auf die dahinter liegende Datenbank, welche alle Fragen und deren Antworten sowie Nutzer und Nutzergruppen beinhalten müsse. Einheitlich als Gruppe hat man sich auf eine opensource Datenbank, welche ohne Installation einsetzbar ist, festgelegt. Die verwendete Datenbank nennt sich MariaDB und ist in dem dazugehörigen Administrationsprogramm „XAMPP“ enthalten. Das Programm XAMPP bietet jedoch nicht nur die Datenbank, sondern auch gleichzeitig das Tool „PHP MyAdmin“ an, welches als Laufzeitumgebung zum öffnen und testen der .php Dateien dient.

## 2.1. Weitere Vorüberlegungen

Zu Beginn des Projektes ist zu überlegen, welche Art von Fragen zu beantworten sind und was dies an technischen Aufwand zur Umsetzung erfordert.

Nach einiger Überlegung konnten wir als Gruppe 3 unterschiedliche Fragetypen mit unterschiedlichen Anforderungen differenzieren. Diese bezeichnen wir im Folgenden als **Schätzfragen, Quizfragen** und **Meinungsfragen** und haben folgende Eigenschaften.

**Schätzfragen oder auch persönliche Fragen** sind Fragen bei denen es schwer ist eine exakte Antwort zu geben, da vielleicht die Lösung auch überhaupt nicht exakt bestimmbar ist, beziehungsweise für jeden anders. Beispielhaft könnte nach der Anzahl der Zuschauer bei einem Sportevent gefragt werden oder dem Gewicht eines Blauwals. Aber auch das Alter oder der Name des Teilnehmers könnte erfragt werden. Der Benutzer bekommt dann ein Textfeld in der er die Lösung eintragen muss. Es gibt also keine vordefinierten Antwortmöglichkeiten. Man kann maximal ein Lösungsbereich vorgeben, dies wäre zum Beispiel mit einem Schieberegler zur Auswahl von Zahlen möglich.  
Da zu erwarten ist, dass jeder Benutzer unterschiedliche Antworten geben wird, ist eine Auswertung nur schwer möglich. Bei Zahlenwerten könnte man einen Mittelwert bilden oder die Antworten Gruppieren und als Gaußkurve aufzeichnen lassen. Dies ist aber alles sehr aufwendig und erst bei wirklich sehr vielen gegebenen Antworten sinnvoll. Bei Textantworten ist eine statistische Auswertung noch schwieriger.

Eine andere Möglichkeit für Schätzfragen, wäre es Lösungsmöglichkeiten vorzugeben, wo wir dann schon bei der nächsten Kategorie von Fragen wären, den Quizfragen.

Bei **Quizfragen** sind verschiedene Antwortmöglichkeiten vorgegeben und die richtige Antwort ist dem Fragesteller bekannt. Somit muss der Nutzer eine von den vorgegebenen Antworten auswählen und es ist direkt nach absenden der Antwort möglich, ihm direktes Feedback zu geben ob die gegebene Antwort richtig oder falsch war.

Diese Art von Fragen haben den spielerischen Charakter eine Quizshow und testen das Wissen der Teilnehmer. Aber auch Schätzfragen können auf diese Art und Weise gestellt werden, indem man Lösungsbereiche als Lösungsmöglichkeit angibt.

Eine dritte Art von Fragen, welche uns in den Sinn gekommen ist, sind die **Meinungsfragen**. Bei Meinungsfragen kann es auch vorgegebene Antwortmöglichkeiten geben, allerdings gibt es hier keine richtigen und falschen Antworten, sondern vielmehr dienen die Fragen dazu, die Meinung der Teilnehmer zu einer Sache zu erfahren. So könnte zum Beispiel die Qualität eines Dienstes auf einer Skala von sehr gut bis hin zu sehr schlecht bewertet werden. Dies liefert Nutzen für den Fragensteller, da er so Feedback von den Teilnehmern erhalten kann.

Für den Anfang wollen wir uns aufgrund des Aufwandes nur für eine Art von Fragestellungen für unser Projekt entscheiden. Aufgrund der einfachen Auswertung, des sofortigen Feedbacks für den Nutzer und der zusätzlichen Möglichkeit auch Schätzfragen damit beantworten lassen zu können, haben wir uns dafür entschieden vorerst nur Quizfragen zu unterstützen.

Für die Zukunft ist aber zu überlegen auch weitere Fragen zu unterstützen. Dazu müsste es auf der Seite zur Erstellung neuer Fragen eine Auswahlmöglichkeit für den Fragetyp geben. Je nachdem welcher Typ gewählt wird, würde dann ein unterschiedliches Formular geladen. Die Datenbank würde dann um ein weiteres Datenfeld „Frage Typ“ erweitert werden und gegebenenfalls müsste eine zusätzliche Tabelle für unterschiedliche Antworttypen angelegt werden.

# 3. Überlegungen während des Projektes

Während der Programmierung und dem Testen unserer Anwendung kam die Frage auf, ob man Fragen und die dazugehörigen Antwortmöglichkeiten bearbeiten können muss.

Zum Beispiel bei Tippfehlern, die beim Frage erstellen auftreten, könnten dadurch ausgebessert werden. Was passiert jedoch, wenn Leute schon auf die Frage geantwortet haben? Bei Kontextänderung der Frage, könnte die Statistik verfälscht werden. Somit wäre eine Änderung nur sinnvoll, solange die Frage noch nicht beantwortet wurde oder wenn es sich wirklich nur um eine einfache Rechtschreibekorrektur handelt. Das lässt sich schwer programmtechnisch überprüfen, ob es sich bei der Änderung der Frage nur um einen Rechtschreibefehler oder einer Inhaltsveränderung handelt. Eine Möglichkeit die uns eingefallen ist wäre eine Änderung nur solange zu zulassen, wie die Frage noch nicht beantwortet wurde.

Letztendlich entschieden wir uns als Gruppe gegen dieses Änderungsfeature, da es mit zu viel Aufwand verbunden ist und der Benutzer die Möglichkeit hat die Frage zu löschen und ggf. wieder neu und richtig zu erstellen.

Jedoch schließen wir die Möglichkeit zur Korrektur von Fragen bis zur ersten Antwort nicht aus und behalten dieses Feature als Could-have für die Zukunft in Errinnerung.

# 4. Test durch unabhängige 3. Person

**-bei falscher Antwort, keine Auflösung was die richtige Lösung war**

-derzeit ist es möglich das mehrere Antwortmöglichkeiten richtig sind, deshalb wird darauf vorerst verzichtet, aber für Zukunft geplant.

**-Popupfenster verhindert laden der Seite und damit weiteres handeln, bevor auf okay geklickt wurde**

->Popupfenster wurden durch grün bzw. rot gefärbte div boxen auf der nächsten Seite ersetzt.



Abbildung 4: Eine farbliche Meldung ersetzt das Popupfenster und ermöglicht ein unterbrechungsfreies Laden der Seite

-**nachdem neue Frage erstellt wurde, fehlt zurück button**  
-> geändert, jetzt Meldung, dass Frage erfolgreich erstellt wurde, Link zu der Frage und Link um eine weiter Frage anzulegen. Das Formular für eine neue Frage wird erstmal nicht mehr angezeigt.

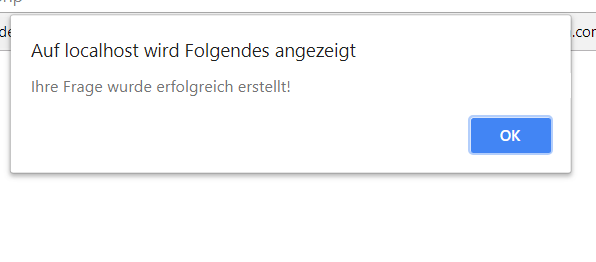


Abbildung 5: Vor der Änderung erschien nur diese Meldung bevor wieder das leere Formular angezeigt wurde



Abbildung 6: Nach den Änderungen

**-wenn man auf den Text der Antwortmöglichkeit klickt, wird immer der oberste Radiobutton ausgewählt**

# 5. Fazit