Inhalt

[Literaturverzeichnis II](#_Toc515802381)

[Abbildungsverzeichnis III](#_Toc515802382)

[1. Allgemeines - 1 -](#_Toc515802383)

[1.1 Aufgabenstellung - 1 -](#_Toc515802384)

[1.2 PHP - 1 -](#_Toc515802385)

[1.3 Structured Query Language - 4 -](#_Toc515802386)

[2. Eigene Ansätze / Vorbereitungen - 6 -](#_Toc515802387)

[2.1. Weitere Vorüberlegungen - 7 -](#_Toc515802388)

[2.3 Datenbank - 9 -](#_Toc515802389)

[3. Überlegungen während des Projektes - 11 -](#_Toc515802390)

[4. Test durch unabhängige 3. Person - 12 -](#_Toc515802391)

[5. Fazit - 14 -](#_Toc515802392)

# Literaturverzeichnis

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

# Abbildungsverzeichnis

[Abbildung 1: PHP-Logo - 2 -](#_Toc515802393)

[Abbildung 2: Architektur - 4 -](#_Toc515802394)

[Abbildung 3: Mind-Map - 6 -](#_Toc515802395)

[Abbildung 4: Tabelle "Users" - 9 -](#_Toc515802396)

[Abbildung 5: Tabelle "user\_group" - 9 -](#_Toc515802397)

[Abbildung 6: Tabelle "questions" - 9 -](#_Toc515802398)

[Abbildung 7: Tabelle "answers" - 9 -](#_Toc515802399)

[Abbildung 8: Tabelle "given\_answers" - 10 -](#_Toc515802400)

[Abbildung 9: Eine farbliche Meldung ersetzt das Popupfenster und ermöglicht ein unterbrechungsfreies Laden der Seite - 12 -](#_Toc515802401)

[Abbildung 10: Nach den Änderungen - 13 -](#_Toc515802402)

# 1. Allgemeines

## 1.1 Aufgabenstellung

Im Rahmen des Studienmoduls Managementtechniken im IT-Projekt (MTITP) galt es eine vorgegebene Aufgabe bzw. Problemstellung mit einer bestimmten Technologie im Bereich der Informatik zu lösen. Dies sollte im Rahmen eines Gruppenprojekts geschehen, bei dem die Arbeitsgruppen von den Studenten selbstständig zusammengestellt wurden. Die Aufgabe erstreckte sich über 2 Theoriesemester und ein Praxissemester.

Eines der zur Wahl stehenden Themen war es, eine Webanwendung für die Beantwortung von Schätzfragen zu erstellen. Hierbei galt es die Aufgabe mit der Technik „Personal Home Page Tools“ oder kurz „PHP“ zu lösen. Zu den Themen der Anderen Gruppen kann an dieser Stelle keine Aussage getroffen werden.

Ziel des Projektes war dabei nicht ein perfektes komplett fehlerfreies Programm zu erstellen, sondern die zu verwendende Technologie, welche sich je Aufgabenstellung unterschied, kennenzulernen. Des Weiteren wurden den Studenten abgesehen von der Aufgabenstellung alle Freiheiten gelassen, die zu erledigende Arbeit selbstständig zu organisieren um ein Gefühl dafür zu erlangen, welche Probleme oder Herausforderungen bei der Planung und Umsetzung eines eigenen Projekts in einer Gruppe auftreten können. Anschließend sollte von der Gruppe ein Fazit zu den gemachten Erfahrungen und den Herausforderungen, die auftraten, zu ziehen und diese in Form dieser Belegarbeit zu Dokumentieren. Um die Erkenntnisse in Bezug auf Herangehensweise und das Erlernte Wissen zur benutzten Technologie mit den Anderen Studenten der Seminargruppe zu Teilen, schließt sich nach Beendigung der Entwicklung und der Abgabe der Belegarbeit eine Präsentation der Anwendung und der genutzten Technologie an.

## 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 und Überlegungen dargestellt, die bei der Lösung der Aufgabenstellung aus Punkt 1.1 vorgenommen wurden.

Zunächst wurde von den Beteiligten eine Mind-Map mit den ersten Ideen und Eindrücken zum Thema erstellt.

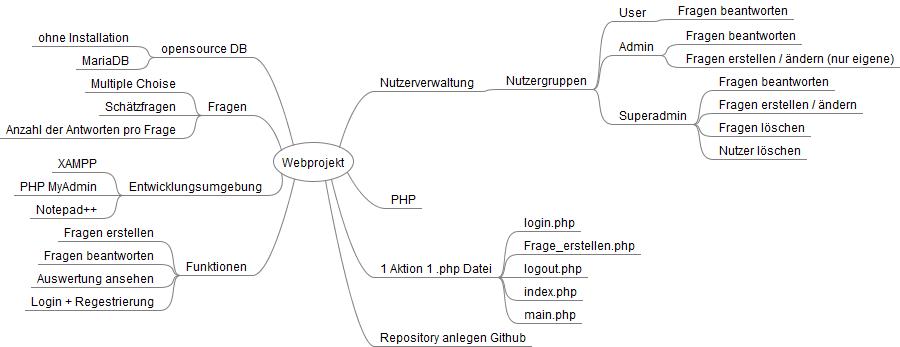


Abbildung 3: Mind-Map

Um diese Aufgabe als Gruppeoptimal lösen zu können, wurde ein Repository benötigt, 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 organisieren und eine Aufgabenteilung ermöglichen zu können.

Als nächstes wurde überlegt, was die Webanwendung später können soll, bzw. was für Funktionen diese haben muss. Ergebnis dieser Überlegungen waren folgende Punkte, die es zu realisieren galt.

* der Nutzer sollte sich einloggen und Registrieren können
* Fragen sollten beliebig erstellt und beantwortet werden können
* eine Auswertung über die beantworteten Fragen sollte dem Nutzer bereitgestellt werden (sowohl über gegebene Antworten, als auch über selbst erstellte Fragen)

Weiterhin braucht es eine Nutzerverwaltung mit Nutzergruppen. Nutzergruppen wären in diesem Fall der allgemeine User, der Admin und der Super-Admin. 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, sowie neue Nutzer anlegen. Die letzteNutzergruppe, der Superadmin, darf zusätzlich zu den Rechten des Admins Fragen und Nutzer löschen.

Als nächstes galt es zu klären, welche Art von Fragen aufgenommen werden sollten und welches Antwortmodell bereitgestellt wird. Hierbei wurde sich entsprechend der Aufgabenstellung für Schätzfragen entschieden, welche über Multiple-Choice-Antwortmöglichkeiten beantwortet werden sollen (je 4 Antworten pro Frage, wobei auch mehrere Antworten korrekt sein können).

Der nächste Gedanke bezog sich auf die dahinterliegende Datenbank, welche alle Fragen und deren Antworten sowie Nutzer und Nutzergruppen beinhalten muss. Hierfür wurde sich auf eine opensource-Datenbank geeinigt, welche ohne Installation einsetzbar ist. 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 „phpMyAdmin“ an, welches als Administrationsoberfläche dient und den Zugriff auf die Datenbank, sowie enthaltene Strukturen und Daten erlaubt.

## 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 wurden 3 unterschiedliche Fragetypen mit unterschiedlichen Anforderungen differenziert. 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 ein Feedback zu geben ob die gegebene Antwort richtig oder falsch war.

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

Aufgrund des Aufwandes, wird nur eine Art von Fragestellung für das Projekt eine Rolle spielen. Angesichts 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 zu überlegen auch weitere Fragen zu unterstützen. Dazu müsste es auf der Seite zur Erstellung neuer Fragen eine Auswahlmöglichkeit für den Fragentyp 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.

## 2.3 Datenbank

In diesem Projekt wird „MariaDB“ als Datenbank benutzt. „MariaDB“ wird mit dem Administrationsprogramm „XAMPP“ mitgeliefert und ist eine opensource Datenbank.

In dieser Datenbank befinden sich insgesamt fünf Tabellen. Die erste Tabelle

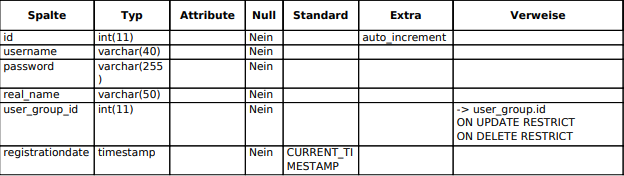


Abbildung 4: Tabelle "Users"

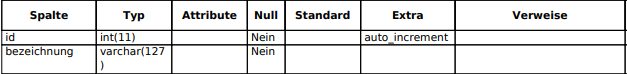


Abbildung 5: Tabelle "user\_group"

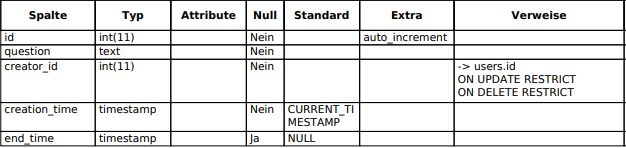


Abbildung 6: Tabelle "questions"

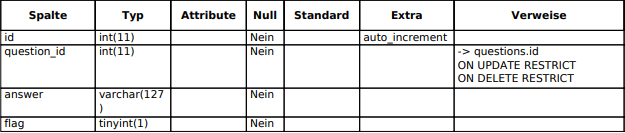


Abbildung 7: Tabelle "answers"

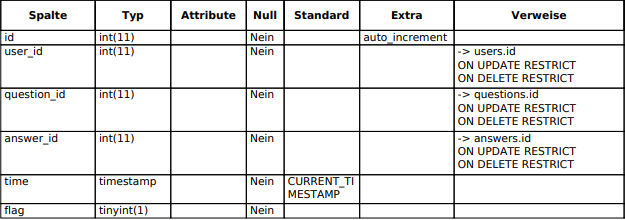


Abbildung 8: Tabelle "given\_answers"

# 3. Überlegungen während des Projektes

Während der Programmierung und dem Testen der Anwendung kam die Frage auf, ob Fragen und die dazugehörigen Antwortmöglichkeiten bearbeitet werden sollen.

Zum Beispiel Tippfehler, welche 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 wäre eine Änderung nur solange zu zulassen, wie die Frage noch nicht beantwortet wurde.

Letztendlich wurde gegen dieses Änderungsfeature entschieden, da es mit zu viel Aufwand verbunden ist. Der Benutzer hat weiterhin die Möglichkeit die Frage zu löschen und ggf. wieder neu und richtig zu erstellen.

Jedoch wird die Möglichkeit zur Korrektur von Fragen bis zur ersten Antwort nicht ausgeschlossen. Dieses Feature wird als Could-have für die Zukunft in Erinnerung gehalten.

# 4. Test durch unabhängige 3. Person

Nach Abschluss der Programmierarbeiten wurde das Projekt von einer unabhängigen Person getestet. Der Grund dafür war eine außenstehende Person Feedback zu dem bisherigen Projektstand geben zu lassen. Anhand des Feedbacks und der Meinung dieser dritten Person lassen sich sogenannte „betriebsblinde Fehler“ erkennen und ggf. ausbessern auch das gesamte Handling des Programmes und die gewählte Umsetzung sollte kritisiert werden.

Die Testperson hat mehrere Kritikpunkte an unserem Projekt feststellen können.

**Bei falscher Antwort erfolgt keine Auflösung der richtigen Antwort.**

Derzeit ist es möglich, dass mehrere Antwortmöglichkeiten richtig sind, deshalb wird darauf vorerst verzichtet. Es ist aber für die Zukunft zusätzlich geplant.

**Die Popupfenster verhindert das Laden der Seite und damit weiteres handeln.**

Popupfenster erschienen nach dem Absenden einer Antwort und nach dem Anlegen einer neuen Frage. Der Nutzer sollte durch sie benachrichtigt werden, dass die Frage erfolgreich erstellt wurde oder es Fehler beim Erstellen der Frage gegeben hat. Ähnlich sieht es beim Beantworten der Fragen aus, nach Abgabe der Antwort wird dem Nutzer mittels Popupfenster mitgeteilt, ob die abgegebene Antwort richtig oder falsch war.



Abbildung 9: Eine farbliche Meldung ersetzt das Popupfenster und ermöglicht ein unterbrechungsfreies Laden der Seite

Die Popupfenster wurden durch grün bzw. rot gefärbte Div-Boxen auf der nächsten Seite ersetzt (siehe Abbildung 4: Eine farbliche Meldung ersetzt das Popupfenster).

**Nachdem eine neue Frage erstellt wurde, fehlte der „Zurück“-Button.**  
Der Nutzer wurde direkt wieder auf das Formular für „neue Frage Anlegen“ weitergeleitet. Jetzt erscheint eine Meldung, dass die Frage erfolgreich erstellt wurde, sowie ein Link zu der neu erstellten Frage und ein Link um eine weiter Frage anzulegen. Das Formular eine neue Frage Anlegen wird erstmal nicht mehr angezeigt (siehe Abbildung 5: Nach den Änderungen).



Abbildung 10: Nach den Änderungen

**Ein Klick auf den Text der Antwortmöglichkeiten führt dazu, dass immer der oberste Radiobutton ausgewählt wird.**

Durch das Klicken auf den Text neben dem Radiobutton führte immer wieder dazu, dass der oberste Radiobutton ausgewählt wird, egal welche Antwortmöglichkeit man gewählt hat. Dieser Fehler im Quellcode wurde mittlerweile behoben und durch das klicken auf den Text neben dem Radiobutton wird die entsprechende Antwortmöglichkeit gewählt.

# 5. Fazit

test