

**Besondere Lernleistung:**

# **Dynamische Webseiten und Content-Management-Systeme**

Wie lässt sich der Datenschutz im digitalen Raum  
realisieren?

Untersuchung am Beispiel der Schulwebsite des  
GutsMuths-Gymnasiums Quedlinburg

**von:**

Bernhard Birnbaum und Noah Wiederhold

**Schule:**

GutsMuths-Gymnasium

**Kurs:**

Informatik

**Schuljahr:**

2019-2020

# Inhaltsverzeichnis

<b>1 Einleitung.....</b>	<b>4</b>
1.1 Hinführung.....	4
1.2 Motivation.....	5
<b>2 Phase 1: Ist-Analyse der aktuelle Schulwebsite des GutsMuths-Gymnasium Quedlinburg...</b>	<b>6</b>
2.1 Ist-Analyse – Funktionen und Usability.....	6
2.1.1 Allgemein.....	6
2.1.2 Front-End-Bereich.....	6
2.1.3 Back-End-Bereich.....	7
2.2 Ist-Analyse – Datenschutz/Datensicherheit.....	11
<b>3 Phase 2: Entwurf einer dynamischen Website mit integriertem CMS für das GutsMuths-Gymnasium Quedlinburg.....</b>	<b>13</b>
3.1 Anforderungen.....	13
3.1.1 Anforderungen – Funktionen und Usability.....	13
3.1.2 Anforderungen - Datenschutz/Datensicherheit.....	14
3.2 Schlussfolgerung.....	16
3.3 Wahl der Programmiersprache.....	16
<b>4 Phase 3: Entwicklung einer dynamischen Website mit integriertem CMS für das GutsMuths-Gymnasium Quedlinburg - Darstellung einzelner Komponenten.....</b>	<b>18</b>
4.1 Entwicklung des Berechtigungssystems und des Log-In-Systems.....	18
4.2 Entwicklung des Stundenplan-Moduls.....	24
4.3 Entwicklung des Kalender-Moduls.....	29
<b>5 Zusammenfassung/Schluss.....</b>	<b>35</b>
<b>6 Zeitaufwandsnachweis.....</b>	<b>36</b>
<b>7 Quellen.....</b>	<b>38</b>
7.1 Literaturverzeichnis.....	38
7.2 Abbildungsverzeichnis.....	40
<b>A Anhang/ Grundlagen.....</b>	<b>45</b>
A.1 Datenschutz allgemein und dessen Relevanz (Quellen: [13], [14], [15], [24] und [25])....	45
A.2 Datenbank Grundlagen (Quellen: [02], [03] und [08]).....	46
A.2.1 Datenbank - Allgemein.....	46
A.2.2 Datenbank – Klärung wichtiger Begriffe.....	46
A.2.2.1 Datenbank – Datensatz.....	46
A.2.2.2 Datenbank – Datenfeld.....	46
A.2.2.3 Datenbank – Query/Abfrage.....	46
A.2.2.4 Datenbank – Redundanz.....	46
A.2.2.5 Datenbank – Primärschlüssel.....	46
A.3 Content Management System (CMS) (Quelle: [01]).....	47
A.3.1 CMS - Allgemein.....	47
A.3.2 CMS - Funktionsweise.....	47
A.3.3 CMS - Aufbau.....	50
A.3.4 CMS – Anwendungsgebiete/Beispiele.....	50

A.4 HTML („Hypertext Markup Language“) (Quellen: [07], [26] und [27]).....	50
A.4.1 HTML – Allgemein.....	50
A.4.2 HTML – Funktionsweise.....	51
A.4.3 HTML – Aufbau.....	53
A.4.4 HTML – Anwendungsgebiete.....	53
A.5 PHP („Hypertext Preprocessor“ bzw. „Personal Home Page Tools“) (Quellen: [08], [10] und [17]).....	53
A.5.1 PHP - Allgemein.....	53
A.5.2 PHP - Funktionsweise.....	53
A.5.3 PHP - Aufbau.....	54
A.5.4 PHP – Anwendungsgebiete.....	55
A.6 CSS („Cascading Style Sheets“) (Quellen: [09], [20], [21] und [22]).....	55
A.6.1 CSS - Allgemein.....	55
A.6.2 CSS - Funktionsweise.....	55
A.6.3 CSS - Aufbau.....	56
A.6.4 CSS - Anwendungsgebiete.....	57
A.7 Javascript (JS) (Quellen: [04], [05] und [06]).....	57
A.7.1 JS – Allgemein.....	57
A.7.2 JS – Funktionsweise.....	58
A.7.3 JS – Aufbau.....	59
A.7.4 JS - Anwendungsgebiete.....	60
A.7.5 JS – jQuery.....	60
A.8 AJAX („Asynchronous JavaScript and XML“) (Quellen: [06] und [16]).....	63
A.8.1 AJAX – Allgemein.....	63
A.8.2 AJAX – Funktionsweise.....	63
A.8.3 AJAX – Aufbau.....	65
A.8.4 AJAX – Anwendungsgebiete.....	65
A.9 SQL (Quelle: [02]).....	65
A.9.1 SQL - Allgemein.....	65
A.9.2 SQL – Funktionsweise.....	65
A.9.3 SQL – Aufbau.....	66
A.9.4 SQL – Anwendungsgebiete.....	67
A.10 MYSQL (Quellen: [03] und [08]).....	67
A.10.1 MYSQL – Allgemein.....	67
A.10.2 MYSQL – Funktionsweise.....	67
A.10.3 MYSQL - Anwendungsgebiete.....	67

# 1 Einleitung

## 1.1 Hinführung

Webprogrammierung ist in der modernen, technisch orientierten Welt eine Aufgabe, welche sich immer mehr Bedürfnissen und Anforderungen, wie z.B. dem Datenschutz, der Datensicherheit oder auch Aspekten der Benutzerfreundlichkeit, anpassen muss. Websites müssen sich dadurch in einem durch Konkurrenz geprägten Wirtschaftsmodell ständig erweitern und immer effizienter werden. Mit einem festgeschriebenen HTML-Grundgerüst (siehe Kap. A.4.3, HTML – Aufbau, S.53) ist es jedoch nur auf Quelltextebene, also durch direktes Verändern des Quellcodes von Websites, möglich die Inhalte und die Darstellung einer Website zu modifizieren. Um Websites funktioneller und effizienter zu gestalten, wurde deshalb der durch HTML sehr stark begrenzte Möglichkeitsspielraum mit Programmiersprachen wie PHP (siehe Kap. A.5, PHP („Hypertext Preprocessor“ bzw. „Personal Home Page Tools“) (Quellen: [08], [10] und [17]), S. 53), CSS (siehe Kap. A.6, CSS („Cascading Style Sheets“) (Quellen: [09], [20], [21] und [22]), S. 55) oder auch JavaScript (siehe Kap. A.7, Javascript (JS) (Quellen: [04], [05] und [06]), S. 57) erweitert.

Diese Sprachen bieten Funktionen, wie z.B. Variablen, mit denen die Anforderungen an Dynamik, Funktionen oder auch Gestaltung realisiert werden können. Dadurch konnten und können vermehrt Aufgaben erleichtert, sowie der „Workflow“<sup>1</sup> im Bereich der Webprogrammierung enorm verbessert werden.

Mit den dynamischen Funktionen ging weiterhin der Wunsch einher, dass Daten nicht nur dynamisch vom Server erzeugt, sondern auch dynamisch gespeichert, sowie ausgelesen werden sollten. Diese Anforderung wurde durch komplexe Bibliotheken erfüllt, welche erstmals die Anbindung an Datenbanksysteme ermöglichten. Mit diesen Systemen lässt sich die Bedeutung der Dynamik erneut erweitern. In den Datenbanken lassen sich Informationen, Daten, Reihenfolgen, und auch ganze Beiträge und Seiten von Websites speichern, welche dann je nach Bedarf vom System aus der Datenbank geladen und dargestellt werden können. Anstelle des Quell-Codes muss fast nur noch die Datenbank gepflegt werden, um aktuelle Inhalte darstellen zu können. Diese systematische Speicherung von Daten wird „Content Management“ genannt, wodurch das Verwaltungssystem der Daten als „Content Management System“ (kurz CMS) bezeichnet wird (siehe Kap. A.3, Content Management System (CMS) (Quelle: [01]), S. 47).

Bezogen auf die Schulwebsite des GutsMuths-Gymnasium Quedlinburg findet diese Form der Informationsbereitstellung z.B. in der Darstellung von einzelnen Beiträgen Anwendung. Anstatt jeden Beitrag statisch<sup>2</sup> als eigene Datei auf dem Server zu speichern oder diese in eine immer größer werdende Hauptdatei einzubinden, werden der Inhalt und die Vorgaben für Struktur und Darstellung eines Beitrages durch bestimmte Mechanismen dynamisch aus einer Datenbank geladen und anschließend dargestellt.

Ein weiterer Aspekt der Dynamischen Websites sind großflächigere Gestaltungsmöglichkeiten und Anzeigevariationen, welche zuerst als Designvorgaben vordefiniert und danach durch die Anwendung je nach Bedarf geladen werden können. Diese Anzeige bezieht sich jedoch nicht nur auf farbliche Aspekte, sondern auch auf Proportionen und Größen, welche sich je nach Endgerät des Benutzers verändern müssen, um Inhalte proportional richtig und gleichzeitig funktional anzuzeigen. Damit ist vor allem die unterschiedliche Darstellung auf mobilen Endgeräten gemeint. Diese Vorgaben basieren im Allgemeinen auf den Stylesheet-Vorgaben der verschiedenen CSS-Dateien einer Website. In diesen ist die Gestaltung der Elemente, sowie deren Verhalten bei verschiedene Rahmenbedingungen, wie z.B. die Fenstergröße des Browsers, festgeschrieben. Hier gibt es bei der aktuellen Schulwebsite einige Defizite.

---

1 Workflow bezeichnet den Arbeitsablauf innerhalb einzelner Prozesse

2 statisch steht in diesem Kontext für eine festgeschriebene Datei, deren Bearbeitung teilweise mit größerem Arbeitsaufwand verbunden ist

## 1.2 Motivation

Das System der aktuellen Schulwebsite ist diesen modernen Anforderungen an eine Website nicht mehr gewachsen. Das zeigt sich z.B. in fehlerhaften Berechtigungssystemen oder an der fehlenden Umsetzung von aktuellen Aspekten des Datenschutzes. Die Gestaltung und Struktur der Website ist außerdem sehr unübersichtlich und die angedachten Funktionen häufig nicht funktionsfähig.

Auf diese Probleme sind wir innerhalb des Informatikunterrichts von unserer Informatiklehrerin Frau Bornschein hingewiesen worden. Nach einigen Gesprächen entwickelte sich nun unsererseits ein gewisser Enthusiasmus, sich der Herausforderung zu stellen und die Schulwebsite von Grund auf zu modernisieren. Da wir nicht auf fertige Systeme, wie z.B. Joomla oder WordPress, zurückgreifen und die gewünschten Funktionen möglichst genau auf die Anforderungen an eine Schulwebsite anpassen wollten, entschieden wir uns dafür, ein komplett neues System auszuarbeiten und dieses im Rahmen einer Besonderen Lernleistung umzusetzen. Der Rahmen dieser Lernleistung umfasst die Entwicklung eines Berechtigungs-, Medien- und Seitensystems, sowie die Umsetzung eines Kalenders, eines Back-End-Bereiches und eines Editors.

Die Entwicklung unterschied sich dabei in einzelne Phasen.

**Phase 1** diente dazu, die existierende Schulwebsite kritisch zu analysieren, Anforderungen und Wünsche für einen neuen Entwurf zu sammeln und ein Konzept zu entwickeln, welches dann in **Phase 2** zunächst zu einer groben Struktur führte. In **Phase 3** wurde abschließend das konzipierte System umgesetzt und die Entstehung dabei schwerpunktartig dokumentiert.

Das Thema reizte uns gerade deswegen sehr, weil die komplexen Zusammenhänge zwischen den einzelnen Aspekten, wie z.B. dem Datenschutz oder der Usability<sup>3</sup>, eine Aufgabe darstellten, der wir uns so noch nicht gewidmet hatten. Außerdem war die Herausforderung, verschiedenste Programmiersprachen, wie z.B. PHP und JavaScript, zu erlernen ein weiterer ausschlaggebender Punkt, warum wir uns für das Thema „Wie lässt sich der Datenschutz im digitalen Raum realisieren“ in Verbindung mit der Entwicklung einer neuen Schulwebsite entschieden haben.

In dieser Arbeit werden die Bestandteile der Website exemplarisch in ihrem Aufbau, ihrer Entwicklung, ihren Anwendungen und ihren Funktionen erläutert, um die Gedankengänge und letztendliche Verwirklichung im Code und in der Funktionsweise der Website nachvollziehbar zu machen.

---

<sup>3</sup> bezeichnet die einfache Bedienbarkeit bzw. die Benutzerfreundlichkeit

## 2 Phase 1: Ist-Analyse der aktuelle Schulwebsite des GutsMuths-Gymnasium Quedlinburg

### 2.1 Ist-Analyse – Funktionen und Usability

#### 2.1.1 Allgemein

Die aktuelle genutzte Schulwebsite bietet auf Grundlage eines frei verfügbaren Content Management Systems viele Funktionen und Möglichkeiten. Das System ist unter dem Namen „Joomla“ bekannt und muss nach der automatischen Installation auf einem Server nur noch konfiguriert werden. Auf der Schulwebsite ist die Version 2.5.28 dieses CMS installiert.

#### 2.1.2 Front-End-Bereich

Im Frontend des Systems können Seiten verschiedener Kategorien und deren Beiträge angezeigt sowie bearbeitet werden. Die Informationsbereitstellung erfolgt jedoch auch über eine Kalenderanwendung und einen extern verlinkten Vertretungsplan. Um die dargestellten Informationen zu bearbeiten muss ein Nutzer sich über einen persönlichen Account authentifizieren und die jeweils benötigten Rechte besitzen. Diese Authentifizierung erfolgt durch ein Login-System, welches verschiedene Nutzergruppen mit unterschiedlichen Rechten unterscheidet. Durch eine fehlerhafte Konfiguration des Systems können diese Gruppen und die damit verbundenen Rechte jedoch nur noch einzeln bzw. gar nicht mehr vergeben werden.



Abbildung 01: Darstellung eines Beitrages der "News"-Seite, welche sich in der Kategorie "Allgemeines" befindet

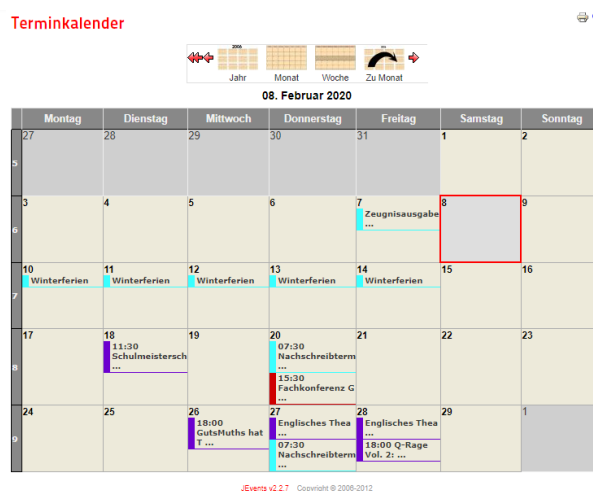


Abbildung 02: Darstellung des Kalenders von JEvents

Neben einer Vielzahl an Beiträgen zum Alltag der Schule und zu verschiedenen formellen Informationen finden sich an verschiedenen Stellen Erwähnungen der verwendeten Software und installierten Erweiterungen. So befindet sich z.B. am unteren Rand der Website die Erwähnung „Powered by Joomla“ oder auch unter der Kalenderanwendung eine Verlinkung des verwendeten Moduls „Jevents v2.2.7“ (siehe Abbildung 02: Darstellung des Kalenders von JEvents).

Die Gestaltung der Website orientiert sich an einer Block-Optik, welche durch ihre kachelartige Struktur einen durchaus modernen Eindruck macht. Die Struktur der Menüs ist jedoch teilweise sehr undurchsichtig, da drei verschiedene Menüs verwendet werden. Die Hauptpunkte werden dabei im Kopfbereich der Website dargestellt und sollen die Navigation zwischen den wichtigsten Seiten vereinfachen.

Diese Vereinfachung findet jedoch nur in einem gewissen Grad statt, da bereits innerhalb des Hauptmenüs Verknüpfungen zu veralteten bzw. irrelevanten Unterseiten existieren. Darunter zählt z. B. die Verknüpfung einer Seite mit Informationen zum „Alumni-Ball“, obwohl dieser 2015 zum letzten Mal stattfand. Die Unterpunkte werden im zweiten Menü in „Allgemeines“ und „Schülerbereich“ unterschieden. Dabei mindern jedoch erneut veraltete Einträge, wie z. B. „E-Learning“, die Übersichtlichkeit und damit die Usability der Webanwendung. Das dritte Menü ist z. B. im Unterpunkt „Projekte“ zu finden und wird für jegliche weiterführende Untergliederung von Seiten verwendet. Zur Gestaltung zählt auch eine Slideshow<sup>4</sup>, welche verschiedene Bilder darstellt, diese jedoch unschön mit einem Watermark<sup>5</sup> des Moduls versehen. Die verwendeten Bilder sind außerdem mehrere Jahre alt und nicht mehr aktuell.

Durch die Authentifizierung eines Mitgliedes des Schülerrates, wird das Schülerrat-Menü angezeigt, welches Neuigkeiten zu Sitzungen, Aktionen und den aktuellen Schülervertretern darstellen soll. Da es jedoch auch hier wenig bzw. keine Aktualisierung von Informationen gibt, ist auch dieser Bereich im Allgemeinen verwaist und wird nicht aktiv genutzt.

Die Website bietet nach der Authentifizierung eines Lehreraccounts einen speziellen „Lehrerbereich“, in welchem z. B. Informationen zu Gesetzen oder auch Medien zu finden sind. Bei ausreichenden Berechtigungen können von hier aus außerdem Beiträge für verschiedene Seiten erstellt werden.

Bezogen auf den Datenschutz besitzt die Website einige kleine Mängel, wie z. B. das Verwenden von Cookies ohne auf deren Einsatz hinzuweisen. Außerdem sind schulinterne Informationen ohne Authentifizierung z. B. im Kalender abrufbar und es gibt keine Angaben über die Umsetzung des Datenschutzes bzw. über deren Verantwortliche.

### 2.1.3 Back-End-Bereich

Der Back-End-Bereich der Schulwebsite ist über die Adresse: „<https://www.gmg.julius-kuehn.de/administrator/>“ verfügbar und bietet nach erfolgreicher Authentifizierung eines Administrators verschiedene Steuerelemente. Dort kann zwischen der Verwaltung von Kategorien, Beiträgen, Medien, Menüs, Benutzer, Benutzergruppen, Plugins und Einstellungen gewählt werden.

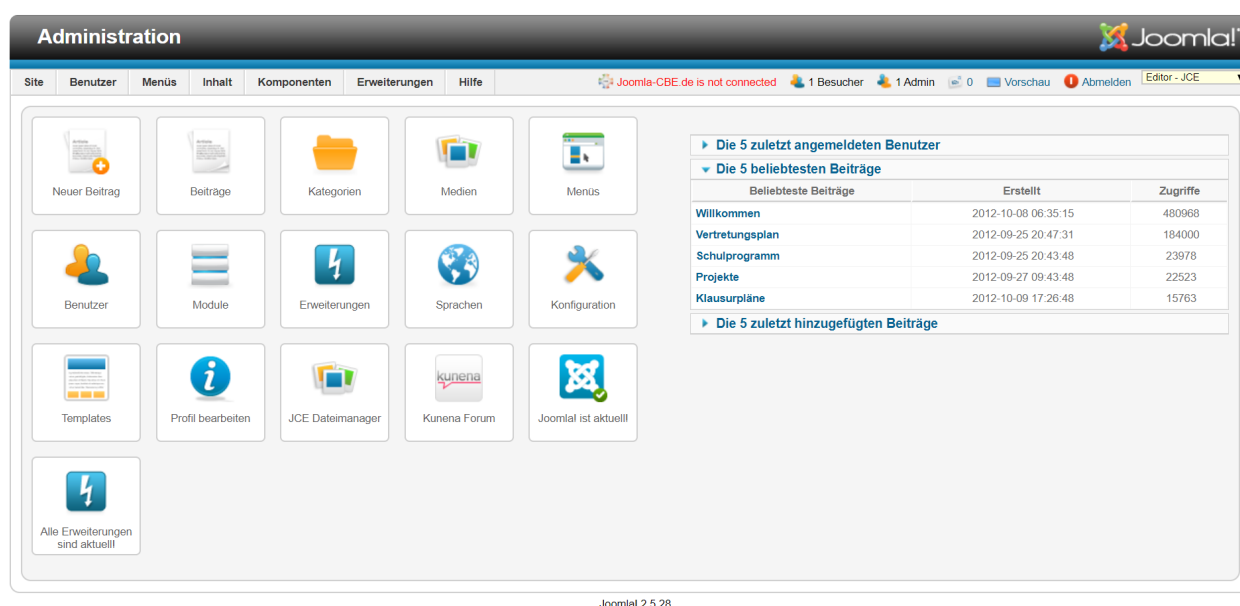


Abbildung 03: Backendbereich mit einzelnen Komponenten

4 Diashow innerhalb eines bestimmten festgelegten Bereichs der Website

5 Darstellung eines Logos bzw. Verweistextes über eigentlichen Inhalten

Die Kategorienverwaltung besteht im Allgemeinen aus einer Tabelle, in welcher die Eigenschaften der einzelnen Kategorien angezeigt werden. Beispielsweise werden dort die Zugriffsebenen oder auch die Titel der Kategorien angezeigt. Gleichzeitig sind verschiedene Buttons vorhanden, welche z. B. die Veröffentlichung von Kategorien umstellen oder auch deren Archivierung einleiten bzw. rückgängig machen können. Weiterhin gibt es die Funktion neue Kategorien zu erstellen und diese mit den gewünschten Eigenschaften zu konfigurieren.

Kategorien: Beiträge									
<div>  Neu          Bearbeiten          Freigeben          Sperrern          Archivieren          Einchecken          Papierkorb          Wiederherstellen          Optionen          Hilfe       </div>									
<div>         Beiträge   <b>Kategorien</b>   Hauptbeiträge       </div>									
<div>         Filter: <input type="text"/> Suchen Zurücksetzen         <div>           - Max. Ebenen wählen - ▼           - Status wählen - ▼           - Zugriffsebene wählen - ▼           - Sprache wählen - ▼         </div> </div>									
<input type="checkbox"/>	Titel	Status	Reihenfolge	Zugriffsebene	Sprache	ID			
<input type="checkbox"/>	Topmenu (Alias: topmenu)	✓	1	Public	Alle	8			
<input type="checkbox"/>	Menu Links (Alias: menu-links)	✓	2	Public	Alle	9			
<input type="checkbox"/>	Lehrerbereich (Alias: lehrerbereich)	✓	3	Public	Alle	10			
<input type="checkbox"/>	Schülerrat (Alias: schuelerrat)	✓	4	Public	Alle	11			
<input type="checkbox"/>	News (Alias: newsfeed)	✓	5	Public	Alle	13			
<input type="checkbox"/>	Projekte (Alias: projekte)	✓	6	Public	Alle	14			
<input type="checkbox"/>	Förderverein (Alias: foerderverein)	✓	7	Public	Alle	15			
<input type="checkbox"/>	Alumni (Alias: alumni)	✓	8	Public	Alle	16			
<input type="checkbox"/>	L.C. (Alias: l-c)	✓	9	Public	Alle	17			

Abbildung 04: Ansicht der Kategorienverwaltung

Aus einer ähnlichen Tabelle besteht auch die Beitrags-/Seitenübersicht. Diese wird jedoch um einen Editor ergänzt, welcher entweder beim Bearbeiten oder beim Erstellen von Beiträgen aufgerufen wird. Im Editor steht auch eine Medien-Funktion zur Verfügung, welche es ermöglicht, z.B. Bilder zu Beiträgen hinzuzufügen bzw. diese auf die Website hochzuladen.

Neu

Bearbeiten

Freigeben

Sperrern

Hauptbeitrag

Archivieren

Einchecken

Papierkorb

Optionen

Hilfe

Beiträge

Kategorien

Hauptbeiträge

Filter:

Suchen

Zurücksetzen

- Status wählen - ▼

- Kategorie wählen - ▼

- Max. Ebenen wählen - ▼

- Zugriffsebene wählen - ▼

- Autor wählen - ▼

- Sprache wählen - ▼

<input type="checkbox"/>	Titel	Status	Hauptbeitrag	Kategorie	Reihenfolge	Zugriffsebene	Autor	Datum	Zugriffe	Sprache	ID
<input type="checkbox"/>	2. Frühlingstreffen vom bis Mai 2013 in Lüneburg (Alias: fruehlingstreffen)		<input type="radio"/>	News L.C.	<div>2</div>	Public	admin_gmg	19.10.2012	293	Alle	54
<input type="checkbox"/>	Aktionen (Alias: aktionen)		<input type="radio"/>	Schülerrat	<div>3</div>	Public	Bornschein	29.01.2014	4137	Alle	111
<input type="checkbox"/>	Aktivitäten (Alias: aktivitaeten)		<input type="radio"/>	Projekte	<div>14</div>	Public	Bornschein	29.11.2012	931	Alle	63
<input type="checkbox"/>	Allgemeine Hinweise zum Vertretungsplan (Alias: allgemeine-hinweise-zum-vertretungsplan)		<input type="radio"/>	Topmenu	<div>22</div>	Public	admin_gmg	09.11.2012	1	Alle	59
<input type="checkbox"/>	Alumni (Alias: alumni)		<input type="radio"/>	Alumni	<div>0</div>	Public	admin_gmg	09.10.2012	13589	Alle	49
<input type="checkbox"/>	ARTintensiv (Alias: ag-kunst)		<input type="radio"/>	Projekte	<div>21</div>	Public	admin_gmg	04.10.2012	3698	Alle	30
<input type="checkbox"/>	Bewertungstabelle_neu (Alias: bewertungstabelle-neu)		<input type="radio"/>	News	<div>7</div>	Lehrer	Bornschein	20.08.2019	0	Alle	328
<input type="checkbox"/>	Cajón-Bau (Alias: cajon-bau)		<input type="radio"/>	Projekte	<div>0</div>	Public	Bornschein	13.09.2016	1566	Alle	219
<input type="checkbox"/>	Chor (Alias: chor)		<input type="radio"/>	Topmenu	<div>17</div>	Public	Bornschein	20.01.2014	9015	Alle	110

Abbildung 05: Ansicht der Seitenübersicht mit Verlinkung zum Editor über den Namen der Seiten, bzw. über die Erstellung einer neuen Seite



### Beitrag bearbeiten

Titel *	Aktionen
Alias	aktionen
Kategorie *	- Schülerrat ▼
Status	Freigegeben ▼
Zugriffsebene	Public ▼
Berechtigungen	Zugriffsrechte setzen
Haupteintrag	Nein ▼
Sprache	Alle ▼
ID	111

Beitragsinhalt


Quedlinburg, 21.12.13

#### Mit Kugeln helfen - Schüler rufen Spendenaktion ins Leben







Weihnachten bedeutet für uns und andere Mitglieder des Schülerrates des GutsMuths-Gymnasiums Quedlinburg, an Andere zu denken und ihnen eine Freude zu bereiten. Als wir erfuhren, dass ein Lehrer unserer Schule Kontakt zum Kinderhospiz Halle der „Björn Schulz STIFTUNG“ hat, begannen wir, eine Spendenaktion zu organisieren. Dieses Hospiz begleitet schwerst- oder unheilbar erkrankte Kinder und unterstützt deren Angehörige. Um diese Aufgaben zu bewältigen, erfordert es intensive menschliche Zuwendung durch die ehrenamtlichen, ausgebildeten Familienbegleiter in Ergänzung zur bestehenden professionellen Pflege- und Schmerztherapie. Wir wollten die Mitarbeiter der Stiftung bei ihrer Arbeit finanziell unterstützen. Unter dem Motto „GutsMuths hilft! Schmückt unseren Weihnachtsbaum für eine gute Tat!“ verkauften wir Christbaumkugeln für je 50 Cent an Schüler und Lehrer. Durch weitere Unterstützung Quedlinburger Sponsoren, unter Anderem von „Parfumerie Flair“, „Schätzkästchen Geschenkartikel“, „Steakhaus St.

- [► Konfigurieren des Editorfensters](#)
- [► Bilder und Links](#)
- [► Metadatenoptionen](#)












In der Menü-Übersicht kann die Konfiguration von verschiedenen Menü-Elementen erfolgen, welche innerhalb einzelner Beiträge bzw. im Grundgerüst der Webanwendung angezeigt werden können. In der Einzelansicht der Menüs können z.B. auch einzelne Menüpunkte temporär entfernt/gesperrt werden, bzw. nur für eine bestimmte Nutzergruppe verfügbar gemacht werden.



# Menüs: Menüs

 Neu
  Bearbeiten
  Löschen
  Wiederherstellen
  Optionen
  Hilfe

[Menüs](#) | [Menüeinträge](#)

	Titel	Anzahl der Menüeinträge			Zum Menü zugeordnete Module	ID 
		# Freigegeben	# Gesperrt	# im Papierkorb		
						
	Topmenu (Menütyp topmenu)	8	1	5	• Topmenu (Public in position-1)	3
	Allgemeines (Menütyp allgemeines)	6	0	0	• Allgemeines (Public in position-7)	4
	Schülerbereich (Menütyp schuelerrat-informiert)	9	0	4	• Schülerbereich (Public in position-7)	5
	Lehrerbereich (Menütyp lehrerbereich)	8	0	60	• Lehrerbereich (Lehrer in position-6)	6
	Schülerrat intern (Menütyp schuelerrat-intern)	1	0	1	• Schülerrat (Schülerrat in position-6)	7
	Projekte (Menütyp menue-rechts)	23	0	3	• Projekte (Public in position-8)	8
	Förderverein (Menütyp foerderverein)	11	0	0	• Förderverein (Public in position-6)	9
	Alumni (Menütyp alumni)	3	1	0	• Alumni (Public in position-6)	10
	CbeCommunity toolbar (Menütyp cbecommunity)	19	0	0	Ein Modul für diesen Menütyp hinzufügen	12

Die Medienverwaltung erfolgt auf der Schulwebsite durch die vorinstallierte Lösung von Joomla bzw. durch ein Modul von „JCE“. Innerhalb dieser beiden Medienverwaltungen werden alle Daten und Ordner, welche sich im Medienverzeichnis des Servers befinden, in zwei Tabellen angezeigt. Die installierte Joomla-Version sortiert Daten dabei standardmäßig nach Namen und die Version von „JCE“ lässt den Benutzer das Sortieren konfigurieren. In beiden Anwendungen werden jedoch, wenn verfügbar, jeweils verschiedene Informationen zu den Dateien angezeigt und man hat auch bei beiden die Möglichkeit Dateien und Ordner zu löschen bzw. neue Dateien hochzuladen oder neue Ordner zu erstellen.

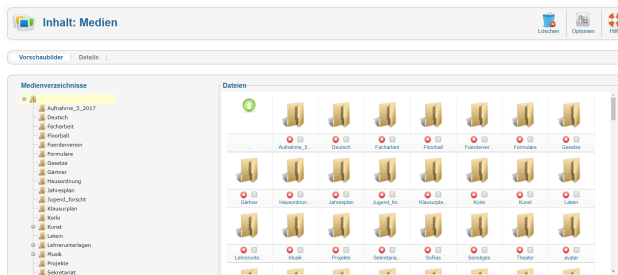


Abbildung 08: Medienverwaltung von Joomla

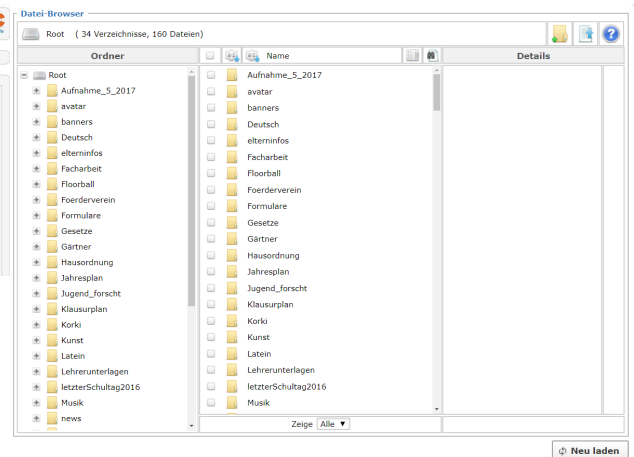


Abbildung 09: Medienverwaltung von "JCE"

Im Unterpunkt „Benutzer“ können alle Benutzerkonten angezeigt, bearbeitet, freigeschaltet oder auch gelöscht werden. Hier werden, z. B. Informationen zu letzten Logins oder auch Registrierdaten angezeigt und es können weitere Nutzer erstellt werden. Funktionen der Rechtevergabe sind Software-seitig theoretisch möglich, jedoch durch eine falsche Konfiguration der dazugehörigen Benutzerkontendatenbank nicht mehr verwendbar. So können z.B. gesperrte Beiträge trotz Adminrechten nur vom Account, welcher diesen gesperrt hat, wieder entsperrt und erst danach durch andere Benutzer bearbeitet werden.



Abbildung 10: Errormeldung nach Interaktion mit gesperrtem Beitrag

In der Erweiterungsverwaltung werden alle installierten Module aufgelistet und können von dort aus aktualisiert und verwaltet werden. Die installierten Module sind jedoch häufig in unterschiedlichen Ausführungen vorhanden bzw. nicht aktiviert und verbrauchen nur unnötig Ressourcen. Die Darstellung der Module ist zudem sehr unübersichtlich und lässt nur theoretisch eine einfache Verwaltung zu.

Ein Beispiel eines ungenutzten Moduls der Website ist das Modul „Kunena Forum“, welches Joomla um die Funktionen einer Forensoftware erweitert. Dieses Modul ist zwar aktiviert, wird jedoch nicht verwendet und ist auch nicht eingerichtet.

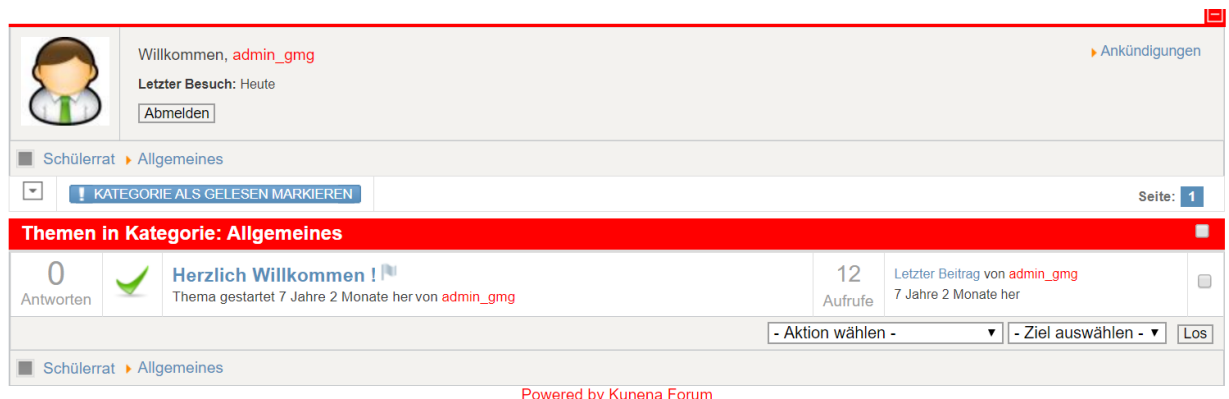


Abbildung 11: Joomla Modul "Kuen Forum" in der Ansicht des Frontend

Im Back-End-Bereich lassen sich auch verschiedene Einstellungen vornehmen. So kann man Sprachpakete für Joomla anwenden oder auch die Meta-Einstellungen, sowie die allgemeinen Einstellungen von Server und System konfigurieren.

## 2.2 Ist-Analyse – Datenschutz/Datensicherheit

Die aktuelle Schulwebsite läuft, wie bereits erwähnt, auf der Version 2.5.28 des CMS „Joomla“. Diese Version unterstützt dabei zwar viele funktionelle Anforderungen, jedoch nicht die neuesten Bestimmungen des Datenschutzes an Schulen.

Die Grundlagen der Datensicherheit sind auf der Website erfüllt. Der Zugriff auf Daten und dadurch auch das Bearbeiten von Daten wird durch ein Login-System eingeschränkt. Dieses verhindert dabei nicht nur den Zugriff, wenn z.B. ein falsches Passwort eingegeben wird, sondern schaltet sich auch zwischen den Aufruf von Seiten mit höheren Rechten, um den Zugriff auf sensible Daten bzw. deren Manipulation zu verhindern.

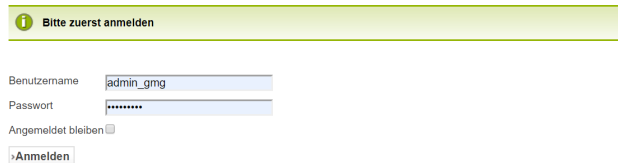


Abbildung 12: Login erforderlich bei Aufruf einer Seite, welche höhere Rechte erfordert ("<https://www.gmg.julius-kuehn.de/index.php/beitrag-erstellen1>")

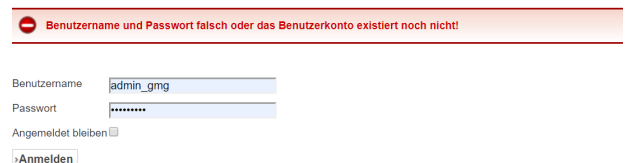


Abbildung 13: fehlgeschlagener Login bei Eingabe eines falschen Passworts

Der Aspekt der Verfügbarkeit ist Hardware-seitig fast durchgängig gegeben, da die Schulwebsite vom Julius-Kühn-Institut gehostet<sup>6</sup> wird und die Server des Instituts durch Fachpersonal überwacht und, falls nötig, auch gewartet werden. Software-seitig ist die Website teilweise instabil und benötigt z.B. beim Laden mehr Zeit und Ressourcen als die neue Website.

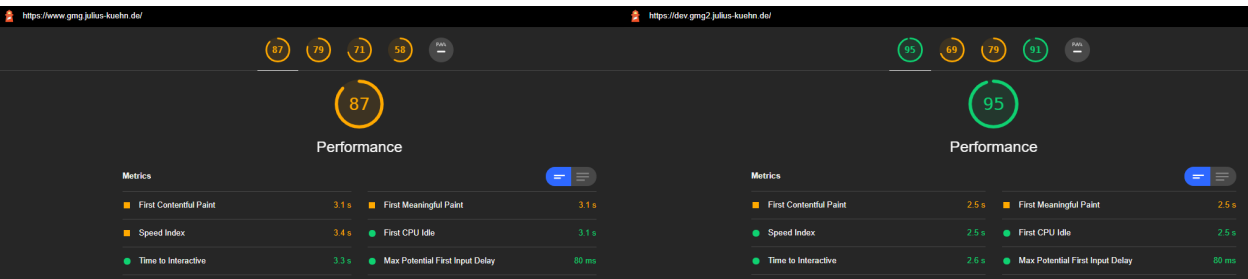


Abbildung 14: Performance Test der aktuellen Schulwebsite mit der Testsoftware Audits

Abbildung 15: Performance Test der neuen Schulwebsite mit der Testsoftware Audits in der Entwicklungsphase

Der Aspekt der Integrität von Daten ist in fast allen Bereichen gegeben. So werden z.B. die Autoren von Beiträgen im Beitragsmenü angezeigt, die Nachvollziehbarkeit von Änderungen ist jedoch nicht gewährleistet.

	Titel	Status	Haupteintrag	Kategorie	Reihenfolge	Zugriffsebene	Autor	Datum	Zugriffe	Sprache	ID
	2. Frühlingstreffen vom bis Mai 2013 in Lüneburg (Alias: fruehlingstreffen)	✓	⊖	News L.C.	2	Public	admin_gmg	19.10.2012	293	Alle	54

Abbildung 16: Eintrag der Beitragsübersicht z.B. mit Angabe des Autors, Veränderungen werden jedoch nicht dokumentiert

6 Bereitstellen eines Servers

Die Aspekte des Datenschutzes, welche eine Schulwebsite erfüllen muss (siehe Kap. A.1 Datenschutz allgemein und dessen Relevanz (Quellen: [13], [14], [15], [24] und [25]) S.45), sind also nicht vollständig erfüllt. Persönliche Daten, wie z.B. die E-Mail-Adresse der Nutzer, und auch sensible Daten, wie z.B. Passwörter werden in einer eigenständigen Datenbank gespeichert und dort teilweise nur verschlüsselt aufbewahrt. Informationen über den Stundenplan der Schüler oder auch über Termine bestimmter Personengruppen sind jedoch frei verfügbar und nicht geschützt. Als Beispiel beinhaltet der Stundenplan sensible Informationen über den Aufenthaltsort der Schüler. Der Zugriff auf diese Informationen muss laut den Anforderungen des Landes Sachsen-Anhalt eigentlich beschränkt werden.

In der technischen Umsetzung ist der Stundenplan aber über den Link „<https://www.gmg.julius-kuehn.de/test/vSindex.html>“ ohne Authentifizierung o.Ä. erreichbar. Dadurch kann der passwortgeschützte Zugriff auf die Vertretungsdaten erfolgen und eine Authentifikation umgangen werden. Der Datenschutz ist also nach heutigen Maßstäben und Anforderungen nicht mehr gegeben bzw. erfüllt.

#### ▪ Vertretungsplan

Achtung: Bitte die Kontaktdaten beim Klassenlehrer erfragen!

Abbildung 17: Link zum Vertretungsplan auf der Schulwebsite

Abbildung 18: Authentifizierung durch Drittanbieter Indiware um auf Vertretungsplan der Schule zuzugreifen

Indiware Vertretungsplan online

Version 6.9.1

Vertretungsplan für: **Dienstag, 14. Januar 2020 (8-Woche)**  
GutsMuths-Gymnasium Quedlinburg  
13.01.2020, 07:53

**Abwesende Lehrer:** Frau Herm (3-6), Frau Jagotzky, Herr Müller  
**Lehrer mit Änderung:** Herr Broede, Herr Burmeister, Frau Cherubim, Frau Dittmann, Frau Herm, Frau Nier-John, Herr Quast, Frau Rummel, Herr Schütze, Frau Zeiske

**Klassen mit Änderung:** 10/2, 10/3, 11/1, 11/2, 11/3, 5/1, 7/1, 8/3, 9/1, 9/3

**Geänderte Aufsichten:**  
11:00: Pause2 - KONVENT -> Frau Nier-John (für Frau Herm)

**Geänderte Unterrichtsstunden:**

Klasse/Kurs	Stunde	Fach	Lehrer	Raum	Info
5/1	5-6	EVA4	Frau Cherubim	009	für EVA4 Frau Zeiske
7/1	1-2	---	---	---	Che Frau Jagotzky fällt aus
7/1	7	---	---	---	Mat Herr Broede fällt aus
8/3	4	Phy	Herr Quast	206Ph	für Mat Herr Burmeister
8/3	7	Frz	Frau Dittmann	108	für Frz Herr Müller
8/3	8	Frz	Frau Dittmann	108	---

www.indiware.de

Abbildung 19: Ansicht des Vertretungsplans bei Eingabe korrekter Benutzerdaten

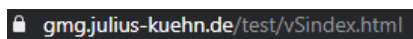


Abbildung 20: Direktlink zum Vertretungsplan ermöglicht das unbemerkte Umgehen der Benutzerdatenüberprüfung

Vertretungsplan für: **Montag, 13. Januar 2020 (8-Woche)**  
GutsMuths-Gymnasium Quedlinburg  
13.01.2020, 07:56

**Abwesende Lehrer:** Frau Böttcher U., Herr Müller, Herr Reilß  
**Abwesende Klassen:** 10/2  
**Lehrer mit Änderung:** Herr Broede, Frau Pichmann, Herr Röhler, Frau Wönsch  
**Klassen mit Änderung:** 10/1, 12/1, 12/2, 12/3, 5/2, 8/3, 9/1, 9/2, 9/3

**Geänderte Unterrichtsstunden:**

Klasse/Kurs	Stunde	Fach	Lehrer	Raum	Info
5/2	5-6	Spa	Herr Röhler	Tu16	für Spa Herr Reilß
5/2	7	---	---	---	Mat Frau Böttcher U. fällt aus
8/3	1-2	---	---	---	Frz Herr Müller fällt aus
9/1-9/3	3-4	Frz	Frau Wönsch	DEU1	für Frz Herr Müller, Aufgaben Fachlehrer
10/1	5-6	Mat	Herr Broede	MAT1	für Mat Frau Böttcher U.
12/1-12/3 sp02	9-10	---	---	---	sp02 Herr Reilß fällt aus

Vertretungsplan für: **Dienstag, 14. Januar 2020 (8-Woche)**  
GutsMuths-Gymnasium Quedlinburg  
13.01.2020, 07:53

**Abwesende Lehrer:** Frau Herm (3-6), Frau Jagotzky, Herr Müller  
**Lehrer mit Änderung:** Herr Broede, Herr Burmeister, Frau Cherubim, Frau Dittmann, Frau Herm, Frau Nier-John, Herr Quast, Frau Rummel, Herr Schütze, Frau Zeiske

**Klassen mit Änderung:** 10/2, 10/3, 11/1, 11/2, 11/3, 5/1, 7/1, 8/3, 9/1, 9/3

**Geänderte Aufsichten:**  
11:00: Pause2 - KONVENT -> Frau Nier-John (für Frau Herm)

**Geänderte Unterrichtsstunden:**

Klasse/Kurs	Stunde	Fach	Lehrer	Raum	Info
5/1	5-6	EVA4	Frau Cherubim	009	für EVA4 Frau Zeiske
7/1	1-2	---	---	---	Che Frau Jagotzky fällt aus
7/1	7	---	---	---	Mat Herr Broede fällt aus
8/3	4	Phy	Herr Quast	206Ph	für Mat Herr Burmeister
8/3	7	Frz	Frau Dittmann	108	für Frz Herr Müller
8/3	8	Frz	Frau Dittmann	108	---
9/1	3-4	Mat	Herr Broede	PHY2	für Che Frau Jagotzky
9/3	5-6	Bo	Frau Rummel	BIO1	Klausur in der Aula
10/2	5-6	Frz	Frau Zeiske	FRZ1	für Frz Herr Müller
10/3	3-4	Frz	Frau Zeiske	KUN1	für Frau Herm
10/3	7-8	Deu	(Frau Zeiske)	FRZ1	selbst, (9), Aufgaben Fachlehrer für zu Hause
11/1-11/3 / 7001-1	1-2	Deu	Herr Schütze	GEO1	Raumänderung

Abbildung 21: Ansicht des Vertretungsplans bei Eingabe des Direktlinks

## **3 Phase 2: Entwurf einer dynamischen Website mit integriertem CMS für das GutsMuths-Gymnasium Quedlinburg**

### **3.1 Anforderungen**

Die Anforderungen an eine sichere Schulwebsite lassen sich in zwei verschiedene Bereiche teilen, die Funktionen der Website und den Bereich des Datenschutzes, sowie der Datensicherheit. Der Aspekt der Funktionen beschäftigt sich mit den sichtbaren Teilen der Webanwendung, wohingegen die Aspekte des Datenschutzes und der Datensicherheit im Hintergrund realisiert werden müssen.

Um die Webanwendung möglichst effizient zu gestalten, soll sich die praktische Entwicklung der Website vor allem auf die notwendigsten Funktionen konzentrieren, um so einen Vorteil zu bereits fertigen Lösungen, wie z.B. Joomla, zu bieten. Dadurch sollen Ladezeiten und Anforderungen an die Hardware verringert und die Übersichtlichkeit, sowie die Verwaltung der Website erleichtert werden.

#### **3.1.1 Anforderungen – Funktionen und Usability**

Unabhängig von sicherheitstechnischen Aspekten muss eine Schulwebsite zahlreichen Anforderungen gerecht werden. Für die effektive Nutzung der Anwendung ist eine durchdachte Struktur und ein einheitlicher Aufbau unerlässlich.

Die Website soll im Allgemeinen in einen öffentlichen Bereich und einen Bereich zum Administrieren der Website eingeteilt werden. Der öffentliche Bereich, der Front-End-Bereich, soll hauptsächlich zum Darstellen von Informationen dienen und nur durch einen Editor bzw. einzelne Funktionen von speziellen Seiten, wie z.B. von einem Kalender, das Bearbeiten von Daten ermöglichen. Jegliche tiefergehende Konfiguration des Website soll im Bereich des Administrators, im Back-End-Bereich, erfolgen.

Die Webanwendung soll für unterschiedlichste Anwender nutzbar sein und den Anforderungen des Datenschutzes entsprechen, weshalb eine Login-Funktion eingebunden werden muss, um einzelne Teile bzw. Funktionen der Website vor unberechtigtem Zugriff zu schützen. Jede Seite der Webanwendung muss demzufolge eigene Berechtigungen erfordern, was ein globales Berechtigungssystem mit Benutzerkonten unabdingbar macht.

Die unübersichtlichen Menüstrukturen der aktuellen Website wollen wir durch ein einfaches Hauptmenü ersetzen, welches im Back-End-Bereich angepasst werden kann und durch eine „Drop-Down-Funktion“<sup>7</sup> die Einordnung von Unterkategorien zu bestimmten Hauptseiten ermöglichen soll.

Um über aktuelle Ereignisse, Projekte, Auszeichnungen, o.Ä. zu informieren soll ein Beitragssystem erstellt werden. Dieses soll durch einzelne Beiträge, welche auf der Website veröffentlicht werden, Informationen bereitstellen. Um diese Beiträge zu erstellen und zu gestalten, ist ein Editor erforderlich, welcher die Einbindung von Text, Bildern, Tabellen und Links ermöglicht. Durch diese Beiträge sollen sich z.B. Eltern über den Ablauf des ersten Schultages oder auch über allgemeine Informationen zum GutsMuths-Gymnasium informieren können.

Mit Hilfe eines elektronischen Stunden-, Klausur- und Vertretungsplans sollen die Schüler außerdem zu Hause ihre Termine und Änderungen des Stundenplans einsehen können. Diese Pläne müssen jedoch im Sinne des Datenschutzes (siehe Kap. A.1 Datenschutz allgemein und dessen Relevanz (Quellen: [13], [14], [15], [24] und [25]), S.45) vor unberechtigtem Zugriff

---

7 Unter „Drop-Down-Funktion“ versteht man das Ausrollen weiterer Menüpunkte durch z.B. Klicken auf einen Erweiterungspfeil oder das Schweben der Maus über einem Menüpunkt im Hauptmenü.

geschützt und die Ansicht auf die jeweils für den Nutzer relevanten Informationen begrenzt werden.

Anwender sollen die Website außerdem auf unterschiedlichen Endgeräten nutzen können, weshalb eine Optimierung und Anpassung des Designs auf Mobil- und Desktop-Geräte erfolgen muss. Es soll ein sogenanntes „Responsive Design“<sup>8</sup> gepflegt werden, dessen Aufgabe die Portabilität der Anwendung ist. Zusammen mit dem Design ist jedoch auch die Geschwindigkeit des Seitenaufbaus maßgebend daran beteiligt, die Webanwendung für den Anwender ansprechend zu machen. Deshalb müssen Dateien, welche zum Design der Website oder zu Beiträgen gehören, komprimiert<sup>9</sup> bzw. ein Dateigrößenlimit beim Einbinden von Bildern, o.Ä. gesetzt und der Code sehr performant<sup>10</sup> sowie ressourcenschonend<sup>11</sup> gestaltet werden.

Mit einem ansprechenden Design in Form von minimalistischen Blockelementen soll die Modernität der Schule betont und die Strukturierung der Webanwendung erfolgen. Der kastenartige Aufbau der Website geht jedoch nicht nur auf den designtechnischen Aspekt, sondern auch auf die Funktionalität zurück. So können z.B. Beiträge, welche in einem eher hochformatigem Schreibfeld erstellt und gestaltet werden, besser und korrekter auf mobilen Geräten dargestellt werden.

Die Farbtöne nehmen dabei eine besondere Stellung ein, um mit bereits vorhandenen Objekten und Veröffentlichungen, wie z.B. dem Jahrbuch des GutsMuths-Gymnasiums, eine einheitliches Auftreten zu erzielen. Die Farben sollen demzufolge auf der Schulfarbe Rot basieren. Angedacht ist ein rotes Menü mit weißem Hintergrund auf den jeweiligen Seiten der Website.

### 3.1.2 Anforderungen - Datenschutz/Datensicherheit

Die nachfolgenden Anforderungen basieren auf den gesetzlichen Vorgaben zum Datenschutz und den Grundlagen der Datensicherheit, welche in Kapitel A.1 Datenschutz allgemein und dessen Relevanz (Quellen: [13], [14], [15], [24] und [25]) S.45 näher erläutert werden.

Zu den grundlegenden Sicherheitsvorkehrungen zählt eine sichere Konfiguration des Webserver. Dateifreigaben und Portfreigaben<sup>12</sup> sollen deswegen entweder deaktiviert, oder Software-seitig geschützt werden. So können z.B. Firewalls die Anfragen, welche an einen bestimmten Port<sup>13</sup> geleitet werden, überprüfen und notfalls verwerfen. Damit kann der Datenverkehr bereits in einem ersten Schritt auf ungewollte Verbindungen oder Daten untersucht und gefiltert werden. Mit der Beschränkung von Dateifreigaben können die ungewollten Zugriffe auf Ordnerstrukturen und dadurch auch auf Dateien verhindert werden. So wird bei einem Serveraufruf wie z.B. „website.de/ordner“ keine Auflistung der Dateien und Unterordner, sondern eine Fehlermeldung angezeigt. Dieser Teil der Anforderungen ist durch die Verwendung der Serverstruktur des JKI bereits gegeben und stellt dadurch zwar eine Anforderung an eine sichere Webanwendung dar, ist aber im Rahmen der praktischen Umsetzung der Lernleistung nicht weiter relevant.

Zusätzliche Sicherheitsvorkehrungen können nur im Code der Webanwendung getroffen werden, da der Schutz und die Sicherheit von Daten eine Kombination aus Hardware-seitigen und

---

8 Unter einem „responsive Design“ versteht man die Anpassung der Website an unterschiedliche Endgeräte und deren spezielle Bildschirmgröße. Diese Anpassung erfolgt dabei in Webanwendungen meist mit Größen der Sprache CSS, welche sich in ihren endgültigen Maßen an der Größe des jeweiligen Bildschirms anpassen.

9 Unter Kompression versteht man das Verkleinern von Daten durch die Verwendung bestimmter Algorithmen. Man unterscheidet zwischen verlustbehafteter und verlustloser Kompression.

10 Als „performant“ wird eine möglichst effiziente und an die Anforderungen angepasste Lösung bezeichnet.

11 Als „ressourcenschonend“ wird eine Lösung bezeichnet, welche möglichst wenig Anforderungen an die Hardware eines Anwenders stellt und auch bei leistungsschwachen Endgeräten flüssig arbeitet.

12 Als Portfreigabe wird das Öffnen einer Schnittstelle für Datenverkehr bezeichnet. Diese Schnittstelle ist jedoch auf einen bestimmten Adressbereich begrenzt.

13 Ein Port ist der Eingang und Ausgang jeglicher Internetverbindungen und dient der Zuordnung von Verbindungen sowie Daten zur jeweiligen Zieladresse.

Software-seitigen Maßnahmen ist. Ein Teil alleine würde den Anforderungen des Datenschutzes nicht gerecht werden können.

Verwaltungsbereiche, wie z.B. ein Admin-Panel<sup>14</sup>, sollen durch eine vorausgehende Berechtigungsüberprüfung, z.B. in Form eines Login-Formulars, vor unberechtigtem Zugriff geschützt werden. Diese Bereiche sind im Allgemeinen enorm mächtig und ermöglichen den Zugriff auf sensible Daten, sowie die Verwaltungswerkzeuge der Anwendung und müssen dadurch gesondert geschützt werden.

Für eine Unterteilung der Anwender in einzelne Gruppen sollen verschiedene allgemeine Rechtegruppen erstellt werden. Diese sollen unterschiedlichen Benutzerkonten jeweils nur den Zugriff auf bestimmte Bereiche der Website bzw. Daten ermöglichen und in Kombination mit Log-In-Formularen den unberechtigten Zugriff auf Daten einschränken. Diese Rechteverteilung soll außerdem Dateien vor Manipulation schützen, da nur auf die jeweils zugeordneten Daten zugegriffen werden kann, wenn z.B. ein einzelnen Benutzer kompromittiert werden sollte.

Einen weiteren Sicherheitsaspekt stellt der Datenaustausch zwischen Benutzer und Server dar. Versendete Daten werden deswegen mit verschiedenen Protokollen unkenntlich gemacht. Ein weit verbreitetes Protokoll für Datentransfer an Server ist dabei das „HTTPS-Protokoll“<sup>15</sup>, mit welchem Daten sicher übertragen werden können. Die Identität von Servern in solch einem Datenaustausch soll durch Zertifikate bestätigt werden, welche den Server als eindeutig kennzeichnen.

Diese Anforderungen sind maßgeblich für die Datenvertraulichkeit und die Datenintegrität verantwortlich, da sie verhindern, dass Daten von Unberechtigten angezeigt bzw. verändert werden und sie die Nachvollziehbarkeit von Datenänderungen ermöglichen.

Mit zunehmendem Funktionsumfang und mit der steigenden Verbreitung von Dynamischen Websites und Webanwendungen, nimmt auch die Anzahl an sicherheitsbedenklichen Fehlern zu. Diese sogenannten „Exploits“ werden von Angreifern ausgenutzt, um sensible Daten auszulesen, zu löschen oder zu manipulieren bzw. um im Namen von anderen Personen zu agieren. So könnten z.B. durch einen Exploit in der Webanwendung einer Schule sensible Informationen über Schüler öffentlich zugänglich gemacht oder der Schulalltag durch die Manipulation der Informationen über ausfallende Stunden durcheinandergebracht werden.

Um diese Fehleranfälligkeit zu reduzieren, ist es nötig die Eingaben der Nutzer auf Schädlichkeit zu untersuchen. Durch Eingabefelder, wie z.B. „`<input>`“, kann ohne solche Vorkehrungen eigener Code ausgeführt werden. Diese Angriffsform wird XSS („Cross Site Scripting“) genannt. Dabei werden Zeichenketten, wie z.B. „`<script>alert(anfällig);</script>`“ in einer aus dem Input generierten Anfrage an den Server und an dessen Quellcode verschickt, welcher den schädlichen Code daraufhin fälschlicherweise ausführt. Um das zu verhindern, müssen Eingaben „escaped“<sup>16</sup> werden. Dies kann z.B. durch das Ersetzen von bestimmten Zeichenabfolgen, wie z.B. „`<script>`“, in Zeichenketten erfolgen.

Der letzte Aspekt der Datensicherheit, die Verfügbarkeit, ist durch effizienten Code und die Vermeidung von ineffizientem Code zu gewährleisten. Um diese Anforderung an die Schulwebsite zu erfüllen, haben wir uns dafür entschieden, keine „All-In-One-Lösung“, also kein vorgefertigtes System, welches nur noch angepasst werden muss, zu verwenden, sondern den Code der Website, sowie den Code des zugehörigen Content-Management-Systems (siehe Kap. A.3 Content Management System (CMS) (Quelle: [01]) S.47) von Grund auf neu zu

---

14 Ein Admin-Panel ist ein speziell Abgegrenzter Bereich einer Software, welcher die Konfiguration und Verwaltung einer Website durch die Bereitstellungen verschiedener Funktionen ermöglicht.

15 Das HTTPS-Protokoll ist ein Verfahren, welches Datenverkehr zwischen Anwender und Server verschlüsselt und die Legitimität des anderen Teilnehmers durch Zertifikate sichert.

16 Das „Escapen“ von Eingaben bezeichnet das Ersetzen von potentiell gefährlichen Zeichenfolgen um z.B. zu verhindern, dass der Rahmen einer Eingabe „`<input></input>`“ durch die Eingabe des Strings „`</input>`“ überwunden wird und ein potentiell schädlicher `<script>`-Tag vor dem zweiten „`</input>`“ eingefügt wird. Dadurch würde der Code im `<script>`-Tag ausgeführt werden und es könnte ein potentielles Sicherheitsrisiko entstehen.

schreiben. Dadurch können wir die Website frei von unnötigem Code, nur mit den nötigen Funktionen, erstellen und sparen uns dadurch Anforderungen an den Server, sowie an den Client und ermöglichen schnellen und konstanten Zugriff auf die Website.

### **3.2 Schlussfolgerung**

Auf Grund der Aktualität von dynamischen Webanwendungen soll der Datenschutz nach der Datenschutzgrundverordnung genauer am Beispiel einer Schulwebsite untersucht werden. Die dabei entstehende Anwendung soll dem „GutsMuths-Gymnasium Quedlinburg“ zur Verfügung gestellt werden und modernen Gestaltungs- sowie Funktionsaspekten entsprechen.

Mit dem Anspruch, all diesen komplexen und untereinander verknüpften Anforderungen an den Datenschutz, die Datensicherheit und an die Usability der Webanwendung zu entsprechen, ergibt sich das Ziel einer hochgradig dynamischen, schnellen, sowie strukturierten Webanwendung mit Anbindung an ein Datenbanksystem, um Daten schnell und sicher abrufen und übertragen zu können.

Durch bereits vorhandene Erfahrungen mit den teilweise sehr fehlerintoleranten „Komplett-Lösungen“ im Bereich dynamischer Webanwendungen, wie z.B. Joomla, soll eine von Grund auf selbst programmierte Webanwendung entstehen, welche an die nötigsten Bedürfnisse einer Schule angepasst werden soll. Der Quellcode soll so durchsichtig wie möglich strukturiert und effizient gestaltet werden, um weiterführende Ergänzungen und Verbesserungen vornehmen zu können oder auch im Rahmen einer weiteren Besonderen Lernleistung den Funktionsrahmen der Anwendung zu erweitern.

Zusammenfassend kann man sagen, dass die Anforderungen an eine sichere und praktische Webanwendung und die mit der technischen Umsetzung verbundenen Risiken in Form von Fehlern sehr hoch sind, eine praktische Umsetzung des Datenschutzes, der Datensicherheit, der Funktionalität und der Gestaltung jedoch eindeutig realisierbar ist.

### **3.3 Wahl der Programmiersprache**

Auf Grundlage der vorhergehenden Erläuterungen zu den Anforderungen an die zu erstellende Webanwendung ergibt sich die Notwendigkeit einer komplexeren Programmiersprache, welche nicht nur durch die Verwendung von Variablen sondern auch durch verfügbare Bibliotheken ein hohes Maß an Dynamik, Funktionen und Sicherheit ermöglicht. Die Sprache soll an die Webentwicklung angepasst und für die Arbeit mit Datenbanken optimiert sein, um die Entwicklung zu erleichtern und das Endergebnis so performant wie möglich zu gestalten.

Durch die Dominanz und Verbreitung von PHP im Umfeld der Webprogrammierung ergibt sich ein breites Spektrum an Funktionen, Bibliotheken und eine leistungsfähige Anbindung an das Datenbanksystem MySQL (siehe Kap. A.5.1 PHP - Allgemein S.53).

Eine andere Möglichkeit der Programmiersprache stellt z.B. Perl dar. Jedoch wurde PHP der Programmiersprache PERL vorgezogen, da diese für den Rahmen einer Schulwebsite unserer Meinung nach eine ausreichende Anzahl an Funktionen bietet und PERL, durch seine Komplexität, eher für höher entwickelte Webanwendung interessant ist. Mit der geringeren Komplexität wird außerdem die weiterführende Wartung durch andere Personen im Code erleichtert und die Ausführungsgeschwindigkeit auf dem Server teilweise erhöht. Mit der Verwendung von PHP geht weiterhin auch die Verwendung von HTML, CSS und JavaScript einher.

Als Datenbanksystem haben wir uns für MySQL entschieden, da die Software kostenlos, mächtig und für unsere Anwendung ausreichend ist. Zwar ist die zeitgleiche Bearbeitung von



Daten in dieser Version nicht möglich, jedoch ist das im Rahmen einer Schulwebsite auch nicht nötig, da keine der benötigten Funktionen mit Echtzeit-Daten realisiert werden soll.

Durch die Zusammenarbeit mit dem Julius-Kühn-Institut wurde uns die Wahl einer geeigneten Serverkonfiguration abgenommen und auch das Erstellen eines Zertifikates für die Website erfolgt durch das Institut. Das JKI stellt uns für die Entwicklung der Schulwebsite einen Apache-Server zur Verfügung, dessen Konfiguration bereits erfolgt ist. Die Wahl eines Apache Webservers ermöglicht uns die effektive Nutzung von kompatiblen Schnittstellen zwischen Datenbank, Webanwendung und Server.

Folgende Lösungen sind zusammengefasst ausgewählt wurden:

- HTML (siehe Kap. A.4 HTML („Hypertext Markup Language“) (Quellen: [07], [26] und [27]) S.50) als strukturgebender Teil der Webanwendung
- PHP (siehe Kap. A.5 PHP („Hypertext Preprocessor“ bzw. „Personal Home Page Tools“) (Quellen: [08], [10] und [17]) S.53) als Dynamisches Element in der sonst weitgehend statischen Struktur von HTML
- CSS (siehe Kap. A.6 CSS („Cascading Style Sheets“) (Quellen: [09], [20], [21] und [22]) S.55) zur Vorgabe der Gestaltung der Webanwendung
- JavaScript (siehe Kap. A.7 Javascript (JS) (Quellen: [04], [05] und [06]) S. 57) als dynamisches Element auf der Ebene von HTML
- MySQL (siehe Kap. A.10 MYSQL (Quellen: [03] und [08]) S.67) als Datenbanksystem zur Verwaltung von Daten
- Apache-Webserver als Grundgerüst jeglicher Webanwendungen

## 4 Phase 3: Entwicklung einer dynamischen Website mit integriertem CMS für das GutsMuths-Gymnasium Quedlinburg - Darstellung einzelner Komponenten

### 4.1 Entwicklung des Berechtigungssystems und des Log-In-Systems

Das Berechtigungssystem basiert im Allgemeinen auf vier verschiedenen Berechtigungsleveln, welche innerhalb der „permission\_levels“-Tabelle festgehalten sind. Es wird in vollständiger Zugriff, Zugriff bis zur Bearbeitung, Zugriff zum Anschauen und in kein Zugriff unterschieden. Diese Hierarchie ermöglicht ein hohes Maß an Dynamik in der Festlegung von einzelnen Berechtigungen.

<input type="checkbox"/> Ändern	id	internal	display
<input type="checkbox"/> bearbeiten	1	ALTER	alles
<input type="checkbox"/> bearbeiten	2	EDIT	bearbeiten
<input type="checkbox"/> bearbeiten	3	VIEW	ansehen
<input type="checkbox"/> bearbeiten	4	NONE	nichts

*Tabelle 1: Berechtigungsebenen innerhalb der "permission\_levels"-Tabelle*

Um die jeweiligen Berechtigungseinstellungen von einzelnen Teilen der Website unterschiedlich einstellen zu können, weist die Website jeder neuen Seite und jedem neuen Kalenderereignis eine neue einzigartige Nummer zu, welche innerhalb der „permissions“-Tabelle hinterlegt und mit ihren jeweiligen Rechten verknüpft wird. Dort wird auch der Besitzer der Seite bzw. des Ereignisses durch seine ID festgehalten, da diese Benutzerkonten stets alle Rechte besitzen sollen, unabhängig ob die jeweilige Benutzergruppe des Besitzers die benötigten Rechte für die verschiedenen Zugriffsarten besitzt. Es ist nicht möglich, einzelnen Benutzern spezielle Rechte für einzelne Seiten oder Ereignisse, zuzuordnen. Eine Differenzierung ist nur über die jeweilige Gruppe des Nutzers möglich, da diese Einteilung nicht gefordert wurde und den Pflegeaufwand der Website enorm erhöht hätte.

Das Berechtigungssystem unterteilt sich für jeden Eintrag in der „permissions“-Tabelle, also jeden einzeln konfigurierten Teil der Website, in die Rechte von 1. nicht angemeldeten Benutzern, 2. angemeldeten Benutzern und zusätzlich dazu in die Berechtigung einer bestimmten Benutzergruppe, welche anhand ihrer ID identifiziert wird. Da keine Hierarchie der einzelnen Benutzergruppen gewünscht wurde, kann jedem Eintrag in dieser Tabelle jeweils nur eine Gruppen mit einem eigenen Berechtigungslevel zugeordnet werden.

<input type="checkbox"/> Ändern	id	owner	public	auth	groupperm	groupid
<input type="checkbox"/> bearbeiten	1	9	VIEW	VIEW	EDIT	5
<input type="checkbox"/> bearbeiten	2	1	VIEW	EDIT	ALTER	5
<input type="checkbox"/> bearbeiten	3	1	VIEW	VIEW	EDIT	1
<input type="checkbox"/> bearbeiten	4	1	NONE	VIEW	EDIT	5

*Tabelle 2: Berechtigungseinstellungen von Seiten und Kalendereinträgen*

Das Anlegen von Berechtigungsdaten für eine neue Seite erfolgt im Code der Website im Bereich des Editors oder innerhalb des Kalenders mit Standardwerten. Diese Werte können später im Editor genauer angepasst werden.

Im Nachfolgenden wird das Zuweisen, Bearbeiten und Überprüfen von Rechten bei Seiten erklärt, da sich die Funktionsweise sehr mit der Funktionsweise des Berechtigungssystems bei Kalender-Ereignissen ähnelt bzw. ineinander integriert ist.

Bei der Erstellung von Seiten wird der eingegebene Seitenname ausgelesen und formatiert um Leerzeichen zu vermeiden, welche nicht in allen Bereichen der Webanwendung unterstützt werden. Diese formatierte Bezeichnung wird zusammen mit dem vom Benutzer übergebenen originalen Titel der Seite, einigen vordefinierten Inhalten und einer einzigartigen Berechtigungs-ID in einen neuen Eintrag der „pages“-Tabelle eingetragen.

```
<?php
$PAGE_NAME=str_replace(" ","_",strtolower(SIMPLIFY_STRING($_GET["func_add"])));

SQL_ADD("pages","title,page,category,source,active,views,permissionid","".
$_GET["func_add"].".".$PAGE_NAME'1,'<p>Neue Seite</p>',0,0,$NEW_PERM_ID");
?>
```

*Quellcode 01: Hinzufügen einer Seite zur "pages"-Tabelle mit vorheriger Formatierung des Bezeichners*

Die Berechtigungs-ID entsteht über die Funktion „PERMSYS\_MAKE\_ENTRY()“, welche innerhalb der Berechtigungstabelle einen neuen Eintrag für die Seite anlegt und mit den Standard-Rechten, einer einzigartigen ID und der an die Funktion übergebenen Gruppen-ID verknüpft.

```
<?php
$NEW_PERM_ID=PERMSYS_MAKE_ENTRY("NONE","VIEW","EDIT",GET_USER_GROUP_ID());
?>
```

*Quellcode 02: Funktionsaufruf um neuen Eintrag in der Berechtigungs-Tabelle zu erstellen und die neue Berechtigungs-ID zu erhalten*

```
<?php
function PERMSYS_MAKE_ENTRY($PUBLIC,$AUTH,$GROUP,$GROUPID) {
    $NEW_ID=SQL_MAXVALUE("permissions","id")["id"]+1;
    SQL_ADD("permissions","id,owner,public,auth,groupperm,groupid","$NEW_ID,".GET_USER_ID().".".$PUBLIC,$AUTH,$GROUP,$GROUPID");
    return $NEW_ID;
}
?>
```

*Quellcode 03: Funktion "PERMSYS\_MAKE\_ENTRY()" legt neue Berechtigungen innerhalb der „permissions“-Tabelle an*

Um die benötigten Rechte von Seiten zu verändern muss diese im Editor aufgerufen werden. Im unteren Bereich des Editors können dann die Berechtigungen konfiguriert werden und mit Hilfe des „Speichern“-Buttons wird die Bearbeitungsfunktion des Editors aufgerufen. Diese Funktion wird mit dem durch eine Benutzerübergabe übergebenen Parameter „func\_save“ ausgelöst.

Spezielle Gruppe:	admin
Spezielle Gruppenberechtigung:	alles
Authentifizierte Nutzerberechtigung:	ansehen
Öffentliche Nutzerberechtigung:	ansehen

Abbildung 22: Konfiguration der Berechtigungseinstellungen einer Seite im Editor

Innerhalb dieser Funktion wird die ID der Benutzergruppe, der jeweiligen Rechteebene der Benutzergruppe, der Rechteebene von angemeldeten Benutzern und der Rechteebene von nicht angemeldeten Benutzern mit Hilfe einer eigenen SQL-Funktion aus den Tabellen „usergroups“ und „permission\_levels“ ausgelesen und in Variablen geladen.

```
<?php
$PAGE_PERM_GROUP_ID=SQL_READ("usergroups","id,name","name",
$_POST["page_perm_group_id"])[ "id"];
$PAGE_PERM_GROUP_LEVEL=SQL_READ("permission_levels","internal,display","display",
$_POST["page_perm_group_level"])[ "internal"];
$PAGE_PERM_AUTH=SQL_READ("permission_levels","internal,display","display",
$_POST["page_perm_auth"])[ "internal"];
$PAGE_PERM_PUBLIC=SQL_READ("permission_levels","internal,display","display",
$_POST["page_perm_public"])[ "internal"];
?>
```

*Quellcode 04: Auslesen der jeweiligen ID der im Editor ausgewählten Eigenschaften um diese später als Verweis zu anderen Tabellen verwenden zu können*

Mit Hilfe dieser Variablen wird im nächsten Schritt der jeweilige Eintrag einer Seite oder eines Kalendereintrags in der „permissions“-Tabelle bearbeitet und die neuen Parameter werden gesetzt. Die Identifikation der Seite erfolgt durch die vom Nutzer übergebene Variable „permid“, welche als Primärschlüssel (siehe Kap. A.2.2.5 Datenbank – Primärschlüssel, S.46) innerhalb der Rechte-Tabelle dient. Das Bearbeiten von Daten wird mit Hilfe einer eigenen Funktion realisiert, welche die benötigten Parameter an den SQL-Server übergibt.

```

<?php
SQL_CHANGE_MULTIPLE("permissions","id",
$_POST["permid"],"public='$PAGE_PERM_PUBLIC',auth='$PAGE_PERM_AUTH',groupper
m='$PAGE_PERM_GROUP_LEVEL',groupid='$PAGE_PERM_GROUP_ID'");
?>

```

*Quellcode 05: Verändern von Berechtigungen eines Eintrages der "permissions"-Tabelle mit Hilfe der vorher ausgelesenen ID's der eingegebenen Optionen*

Das Überprüfen der erforderlichen Rechte eines Benutzers für eine normale Seite erfolgt durch ein Verteilerelement von PHP, welches anhand des jeweiligen Berechtigungslevels des Nutzers, bezogen auf eine bestimmte Seite, in vier verschiedene Fälle unterteilt. Diese Unterteilung erfolgt durch den „switch“-Befehl von PHP, welcher eine Fallunterscheidung in unterschiedliche Fälle einer Bedingung ermöglicht. Das Berechtigungslevel des Nutzers wird durch die Funktion „PERMSYS\_GET\_PERM\_FORUSER()“ ausgelesen. Dabei wird die Berechtigungs-ID der Seite genutzt, um die Berechtigungseinstellungen der Seite zu erhalten.

Sollte der Nutzer ein Admin bzw. der Besitzer der Seite sein, wird die Funktion vorzeitig beendet und die höchste Berechtigungsstufe zurückgegeben. Andernfalls werden, falls die Gruppe des Benutzers der eingestellten Gruppe entspricht, zuerst die Rechte der jeweiligen Gruppe des Benutzers mit der für die Seite eingestellten Berechtigung von Gruppen verknüpft. Diese Berechtigung wird als Zahl gespeichert (siehe „permission\_levels“-Tabelle). Die Berechtigungen von authentifizierten Nutzern und von nicht angemeldeten Nutzern werden auch jeweils als Zahl gespeichert und in einer anschließenden Rückgabe wird der kleinste Wert aus jeglichen Berechtigungen ausgewählt, da dies der höchsten Berechtigung entspricht, die der Nutzer im Zusammenhang mit der Seite besitzt. Mit diesem Wert wird anschließend die jeweils zur Berechtigung passende Ansicht realisiert. Zum Beispiel wird die Ansicht der Berechtigungslevel ALTER und EDIT um einen Bearbeitungsbutton ergänzt oder es wird, falls der Nutzer keine Rechte im Zusammenhang mit der Seite besitzt, eine Fehlermeldung angezeigt und die Seite nicht dargestellt.

```

<?php
switch(PERMSYS_GET_PERM_FOR_USER($SQL_PAGE["permissionid"])){
    case "ALTER":
        $CUSTOM_PAGE->ADD_SOURCE($EDITBTN->GET_SOURCE());
        $CUSTOM_PAGE->ADD_SOURCE($SQL_PAGE["source"]);
        PAGE_ADD_VIEW($SQL_PAGE["page"]);
        break;
    case "EDIT":
        $CUSTOM_PAGE->ADD_SOURCE($EDITBTN->GET_SOURCE());
        $CUSTOM_PAGE->ADD_SOURCE($SQL_PAGE["source"]);
        PAGE_ADD_VIEW($SQL_PAGE["page"]);
        break;
    case "VIEW":
        if($SQL_PAGE["active"] == 1){
            $CUSTOM_PAGE->ADD_SOURCE($SQL_PAGE["source"]);
            PAGE_ADD_VIEW($SQL_PAGE["page"]);
        } else {
            $CUSTOM_PAGE->ADD_SOURCE("<h1>Fehler</h1>");
            $CUSTOM_PAGE->ADD_SOURCE("<p class=\"centered\">Die angeforderte Seite
ist aktuell gesperrt.<br>Vielleicht kann die <a href=\"?p=home\">Startseite</a>
weiterhelfen.</p>");
            LOGGING_PUSH("WARNING","Es wurde versucht, eine gesperrte Seite aufzurufen
(".$SQL_PAGE["page"]." ( ".$SQL_PAGE["id"].").");
        }
        break;
    case "NONE":
        $CUSTOM_PAGE->ADD_SOURCE($ACCESS_DENIED->GET_SOURCE());
        break;
    default:
        $CUSTOM_PAGE->ADD_SOURCE("<h1>Fehler</h1>");
        $CUSTOM_PAGE->ADD_SOURCE("<p class=\"centered\">Die angeforderte Seite
kann aufgrund eines Fehlers im Berechtigungssystem aktuell nicht angezeigt
werden.<br>Vielleicht kann die <a href=\"?p=home\">Startseite</a> weiterhelfen.</p>");
        LOGGING_PUSH("WARNING","Es ist möglich, dass ein interner Fehler beim
Aufrufen der Seite ".$SQL_PAGE["page"]." ( ".$SQL_PAGE["id"].")' im Berechtigungssystem
ausgelöst wurde.");
        break;
}
?>

```

*Quellcode 06: Fallunterscheidung der Berechtigungen durch das zurückgegebene Berechtigungslevel der Funktion "PERMSYS\_GET\_PERM\_FOR\_USER()"*

```

<?php
function PERMSYS_GET_PERM_FOR_USER($PERMID){
    $SQL_REQUEST=SQL_READ("permissions","*", "id", $PERMID);
    $PERM_OWNER=$SQL_REQUEST["owner"];
    $PERM_PUBLIC=$SQL_REQUEST["public"];
    $PERM_AUTH=$SQL_REQUEST["auth"];
    $PERM_GROUP=$SQL_REQUEST["groupperm"];
    $PERM_GROUP_ID=$SQL_REQUEST["groupid"];
    if(LOGGED_IN()){
        if(GET_USER_ID() == $PERM_OWNER){
            return "ALTER";
        }elseif(IS_USER_ADMIN()){
            return "ALTER";
        }else{
            $TMP_PERM;
            $USERGROUP=GET_USER_GROUP_ID();
            if($USERGROUP==$PERM_GROUP_ID){
                $TMP_PERM=$PERM_GROUP;
            }else{
                $TMP_PERM="NONE";
            }

            $PERM_PUBLIC_VALUE =
SQL_READ("permission_levels","id,internal","internal",$PERM_PUBLIC)["id"];
            $PERM_AUTH_VALUE =
SQL_READ("permission_levels","id,internal","internal",$PERM_AUTH)["id"];
            $PERM_GROUP_VALUE =
SQL_READ("permission_levels","id,internal","internal",$TMP_PERM)["id"];
            return
SQL_READ("permission_levels","id,internal","id",min($PERM_PUBLIC_VALUE,
$PERM_AUTH_VALUE,$PERM_GROUP_VALUE))["internal"];
        }
    }else{
        return $PERM_PUBLIC;
    }
}
?>

```

*Quellcode 07: Funktion der Website, welche das maximale Berechtigungslevel einer Nutzers im Zusammenhang mit einer Seite zurückgibt*

## 4.2 Entwicklung des Stundenplan-Moduls

Der Stundenplan des Front-End-Bereichs unterteilt sich in zwei Hauptbestandteile, die Anzeige des Stundenplans und die Konfigurationsseite, in welcher der Nutzer seine jeweilige Klasse und die gewählten Kurse einstellen kann. Das System basiert auf einer durch einen Drittanbieterprogramm generierten XML-Datei, welche die kompletten Stundenplandaten für jede Klassen- und Kurskonfiguration inklusive der Vertretungsdaten beinhaltet. Diese wird ausgelesen und jeweils auf den Nutzer und seine Einstellungen angepasst dargestellt.

Im ersten Schritt wird geprüft, ob für den angemeldeten Benutzer Daten über seine Klassen- und Kurswahl vorliegen. Falls diese nicht vorliegen wird der Stundenplan ohne einen Filter angezeigt.

```
<?php
$SQL_ROW = SQL_READ("schedule", "id,user,cl,co", "user", GET_USER_ID());
if($SQL_ROW) {
    $COURSES = $SQL_ROW["co"];
    if($COURSES == "") {
        $COURSES = "keine";
    }
    ...
}
?>
```

*Quellcode 08: Funktion um Konfigurationsdaten zu überprüfen*

Anschließend werden, je nach Nutzerberechtigung, zuerst die entsprechenden Buttons zum Anzeigen des Stundenplans sowie zum Upload einer neuen Stundenplan-XML-Datei eingebunden.



*Abbildung 23: Buttons des Vertretungsplans*

Da die Stundenplananzeige und auch die Ausgabe der Daten im HTML-Format zum größten Teil durch JavaScript in einer externen Datei realisiert wurden, wird anschließend nur noch der zur Anzeige benötigte HTML „table“-Tag mit der ID „stundenplan“ und die angesprochene JS-Datei eingebunden.

```
<?php
$CUSTOM_PAGE->ADD_SOURCE("<h3 style='display: none;'"
id="timestamp"></h3><table style="display: none;" id="stundenplan"></table>");
$CUSTOM_PAGE->ADD_SOURCE("<script language='javascript'" type="text/javascript"
src="javascript/stundenplan.js"></script>");
?>
```

*Quellcode 09: Erstellen von groben Strukturen des Vertretungsplans*



Durch das Anklicken einer der „Plan“-Buttons wird die JavaScript-Funktion „getScheduleData(date, cl, co)“ aufgerufen, wobei „date“ das angeforderte Datum, „cl“ die Klasse und „co“ die Kurse darstellt. Die XML-Datei, insofern diese auf dem Server vorhanden ist, wird dann durch einen AJAX-Aufruf ausgelesen und an die Funktion „applySchedule(xmlData, cl, co)“ weitergegeben.

```
function getScheduleData(date, cl, co = ""){
    $("#timestamp").fadeOut(250);
    $("#stundenplan").fadeOut(250, function() {
        var ajaxUri = "media/stundenplan/PlanKI" + date + ".xml";
        $.ajax({
            cache: false,
            url: ajaxUri,
            success: function(xmlData) { applySchedule(xmlData, cl, co); },
            error: function(x,y,z) {
                if(z == "Not Found") {
                    var d_day = date.substr(6, 2);
                    var d_month = date.substr(4, 2);
                    var d_year = date.substr(0, 4);
                    dpBoxMessage("Plan nicht vorhanden", "Der Stundenplan f&uuml;r  
den " + d_day + "." + d_month + "." + d_year + " ist (noch) nicht vorhanden.");
                } else {
                    dpBoxMessage("Fehler beim Laden", "Der Stundenplan konnte nicht  
geladen werden. Versuchen Sie es sp&auml;ter erneut. Weitere Informationen: " + z + ".");
                }
            }
        });
    });
}
```

*Quellcode 10: Laden der Vertretungsdaten über eine Ajax-Anfrage und Weiterleitung zur "applySchedule"-Funktion*

In der „applyschedule“-Funktion findet nun das eigentliche XML-Parsing statt, welches maßgeblich durch die gegebene Datenstruktur beeinflusst wird. Ein XML-Tag wird zuerst durch die jQuery-Funktion „xml.find(tag)“ selektiert. Im Fall, dass mehrere XML-Tags auf derselben Ebene vorhanden sind, hilft die jQ-Funktion „xml.find(tag).each( function() {...} )“, um jedes Element zu durchlaufen.

Innerhalb dieser Schleife fungiert der „this“-Zeiger als der aktuell selektierte XML-Tag, wodurch auf die jeweiligen Child-Elemente zugegriffen werden kann. Der Inhalt eines XML-Tags wird anschließend direkt durch „xml.find(tag).text()“ ausgegeben und es kann direkt auf ein XML-Attribut über „xml.find(tag).attr(attribute)“ zugegriffen werden.

```

parsedXml.find("Kl").each(function() {
    var xmlCl = $(this);
    if(xmlCl.find("Kurz").text() == cl) {
        xmlCl.find("Pl").find("Std").each(function() {
            var planHour = $(this).find("St").text();
            var planSubject = $(this).find("Fa").text();
            var planSubjectChanged = $(this).find("Fa").attr("FaAe");
            var planTeacher = $(this).find("Le").text();
            var planTeacherChanged = $(this).find("Le").attr("LeAe");
            var planRoom = $(this).find("Ra").text();
            var planRoomChanged = $(this).find("Ra").attr("RaAe");
            var planInformation = $(this).find("If").text();
        });
    }
});

```

*Quellcode 11: schematisches Auslesen der XML-Datei und Einfügen in Variablen*

Insofern keine Kurskonfiguration vorliegt, wird direkt die Funktion „appendScheduleLine(planSubjectChanged, planTeacherChanged, planRoomChanged, planHour, planSubject, planTeacher, planRoom, planInformation)“ aufgerufen, welche Unterrichtsstunden direkt ausgibt. Jedoch im Falle einer Kurswahl des Nutzers werden zunächst alle existierenden Kurse bzw. Fächer aus der XML-Datei ausgelesen und mit den vom Nutzer konfigurierten verknüpft, wodurch eine komplette Liste, bestehend aus Fächern im Klassenverband, sowie der gewählten Kurse, entsteht. Dadurch können nun die Stundenplandaten des angeforderten Tages und der angeforderten Klasse mit den vom Nutzer gewählten Kursen abgeglichen werden, um nur die Fächer und Kurse anzuzeigen, welche für den Schüler relevant sind, d.h. dass auch nur in diesem Fall jeweils die „appendScheduleLine“-Funktion aufgerufen wird.

```

if(co == "") {
    appendScheduleLine(planSubjectChanged, planTeacherChanged, planRoomChanged,
planHour, planSubject, planTeacher, planRoom, planInformation);
} else {
    var clst = "";
    xmlCl.find("Unterricht").find("Ue").each(function() {
        clst += $(this).find("UeNr").attr("UeFa") + ";";
    });
    clst += co;
    for(var i = 0; i < clst.split(";").length; i++) {
        var planCurrentSubject = clst.split(";")[i];
        if(planSubject == planCurrentSubject ||
planInformation.startsWith(planCurrentSubject)) {
            appendScheduleLine(planSubjectChanged, planTeacherChanged,
planRoomChanged, planHour, planSubject, planTeacher, planRoom, planInformation);
        }
    }
}
}

```

*Quellcode 12: Erstellen einer Liste mit für den Schüler relevanten bzw. allen Informationen*

Nun erfolgt abschließend die Ausgabe der Daten im HTML-Format ebenfalls durch die Funktion „appendScheduleLine(planSubjectChanged, planTeacherChanged, planRoomChanged, planHour, planSubject, planTeacher, planRoom, planInformation)“. Dabei werden Zellen mit Änderungen rot eingefärbt, um Abweichungen vom normalen Stundenplan besser zu visualisieren. Diese Tabellenzeilen werden im letzten Schritt in die HTML-Tabelle „stundenplan“ eingefügt, wodurch der Stundenplan entsteht.

```
function appendScheduleLine(planSubjectChanged, planTeacherChanged, planRoomChanged,
planHour, planSubject, planTeacher, planRoom, planInformation) {
    if(planSubjectChanged) {
        appendSchedule("<tr><td>" + planHour + "</td><td class='marked'>" + planSubject +
"</td><td>" + planTeacher + "</td><td>" + planRoom + "</td><td>" + planInformation +
"</td></tr>");
    } else if(planTeacherChanged) {
        appendSchedule("<tr><td>" + planHour + "</td><td>" + planSubject + "</td><td
class='marked'>" + planTeacher + "</td><td>" + planRoom + "</td><td>" + planInformation +
"</td></tr>");
    } else if(planRoomChanged) {
        appendSchedule("<tr><td>" + planHour + "</td><td>" + planSubject + "</td><td>" +
planTeacher + "</td><td class='marked'>" + planRoom + "</td><td>" + planInformation +
"</td></tr>");
    } else {
        appendSchedule("<tr><td>" + planHour + "</td><td>" + planSubject + "</td><td>" +
planTeacher + "</td><td>" + planRoom + "</td></tr>");
    }
}
```

*Quellcode 13: Ausgabe der Daten des XML-Dokuments als Tabellenzeilen*

Neben der Darstellung des Vertretungsplans können auch die jeweiligen Klassen und Kurse eines Schülers in einer entsprechenden Konfigurationsseite eingestellt werden. Bei der Klassenauswahl handelt es sich um ein simples „select“-Element, welches mit Hilfe von zwei ineinander verschachtelten Schleifen befüllt wird.

```
<?php
$CLASS_LIST = array();
$c = 0;
for($i = 5; $i <= 12; $i++) {
    for($j = 1; $j <= 4; $j++) {
        $CLASS_LIST[$c] = $i . "/" . $j;
        $c++;
    }
}
$CLASS_LIST_STR = implode("|", $CLASS_LIST);
$CUSTOM_FORM->ADD_INPUT_SELECT("Klasse", "cl", $CLASS_LIST_STR,
$CLASS_LIST_STR);
?>
```

*Quellcode 14: Ausgabe aller möglichen Klassen innerhalb eines "select"-Elements*

Im Gegensatz dazu wird bei der Kurswahl zunächst eine Liste mit allen Kursen als Grundlage verwendet, aus denen dann Grundkurse (Kleinbuchstaben) und Leistungskurse (Großbuchstaben) generiert werden. Dieses Array wird dem Nutzer anschließend in einem benutzerdefinierten „checkbox“-Element ausgegeben.

```
<?php
$SUBJECT_LIST = explode(";", "bio;che;eng;eth;evr;geo;inf;kun;lat;mus;phy;rdk;soz");
$CUSTOM_FORM->ADD_CUSTOMSOURCE("<label>Kursauswahl:</label><br>");
for($i = 0; $i < count($SUBJECT_LIST); $i++) {
    for($j = 1; $j <= 3; $j++) {
        $COURSE_NAME_GK = ucfirst($SUBJECT_LIST[$i])." - Kurs ".$j." - Grundkurs";
        $COURSE_NAME_LK = ucfirst($SUBJECT_LIST[$i])." - Kurs ".$j." - Leistungskurs";
        $COURSE_SHORT_GK = strtolower($SUBJECT_LIST[$i].$j);
        $COURSE_SHORT_LK = strtoupper($SUBJECT_LIST[$i].$j);
        $CUSTOM_FORM->ADD_INPUT_CHECKBOXARRAY($COURSE_NAME_GK,
        "colistitems", $COURSE_SHORT_GK);
        $CUSTOM_FORM->ADD_INPUT_CHECKBOXARRAY($COURSE_NAME_LK,
        "colistitems", $COURSE_SHORT_LK);
    }
}
$CUSTOM_FORM->ADD_CUSTOMSOURCE("<label>gew&auml;hlte Kurse:</label>");
$SQL_ROW = SQL_READ("schedule", "id,user,co", "user", GET_USER_ID());
if($SQL_ROW) {
    $CUSTOM_FORM->ADD_CUSTOMSOURCE("<input    readonly    type=\"text\"
    name=\"co\" id=\"colist\" value=\"\" . $SQL_ROW[\"co\"] . \"\"/>");
} else {
    $CUSTOM_FORM->ADD_CUSTOMSOURCE("<input    readonly    type=\"text\"
    name=\"co\" id=\"colist\"/>");
}
?>
```

*Quellcode 15: Ausgabe aller möglichen Kurse innerhalb eines benutzerdefinierten "checkbox"-Elements*

Zusätzlich werden alle ausgewählten Kurse durch ein JavaScript-Event in eine Liste im CSV-Format eingetragen, um diese dann später in die Datenbank einzutragen und damit die Auswahl des Nutzers zu speichern.

```

function appendCourse(co) {
    if($("#colist").val() == "") {
        $("#colist").val($("#colist").val() + co);
    } else {
        $("#colist").val($("#colist").val() + ";" + co);
    }
}

$("#colistitems:checked").each(function() {
    appendCourse($(this).val());
});

```

*Quellcode 16: Funktion um gewählte Kurs zu strukturieren*

Um die Benutzerfreundlichkeit zu steigern, werden bereits ausgewählten Kurse dem Nutzer auch dementsprechend angezeigt. Diese Anzeige wird dabei durch diese Funktion realisiert:

```

$("#colistitems").each(function() {
    var courseArray = courseList.split(";");
    for(var i = 0; i < courseArray.length; i++) {
        if($(this).val() == courseArray[i]) {
            $(this).attr("checked", true);
        }
    }
});

```

*Quellcode 17: Funktion um bereits Ausgewählte Kurse hervorzuheben*

### 4.3 Entwicklung des Kalender-Moduls

Der Kalender der Schulwebsite fällt unter die Kategorie der speziellen Seiten, weshalb vor dessen Initialisierung eine Überprüfung der erforderlichen Rechte des Benutzers erfolgen muss. Wenn diese Rechte vorhanden sind, wird der Kalender geladen, ansonsten wird eine Fehlermeldung angezeigt.

Um die Zeit- und Datumsfunktionen von PHP verwenden zu können ohne die Werte der standardmäßigen Zeitzone umwandeln zu müssen, wird die Zeitzone in Europa/Berlin geändert.

```

<?php
date_default_timezone_set("Europa/Berlin");
?>

```

*Quellcode 18: Standardwert der Zeitzone verändern mit Hilfe einer Funktion von PHP*

Die Werte für den anzuzeigenden Monat und das Jahr werden entweder über die vom Nutzer übergebenen Parameter m und y festgelegt oder sie entsprechen den aktuellen Werten. Um die aktuellen Werte zu ermitteln, wurde die „date()“-Funktion von PHP genutzt, welche unterschiedlichste Zeitwerte ausgeben kann.

```
<?php
$YEAR;
if(isset($_GET["y"])){
    $YEAR=$_GET["y"];
} else {
    $YEAR=date("Y",time());
}
?>
```

*Quellcode 19: Festlegen des anzuzeigenden Jahres entweder über den übermittelten Parameter oder über das aktuelle Jahr*

Im Nachfolgenden soll das Anzeigen von Ereignissen sowie das Erstellen, Bearbeiten und Löschen dieser Daten mit Hilfe von Code-Ausschnitten erklärt werden.

Im Allgemeinen wird zuerst die Navigation des Kalenders erstellt, die mit verschiedenen Buttons realisiert wurde, die jeweils zugeordnete Parameter übergeben und damit den anzuzeigenden Zeitraum ändern bzw. Funktionen zum Hinzufügen von Ereignissen ausführen.

```
<?php
$CALENDAR_BUTTONS->ADD_BUTTON_NAVIGATION("<i class='fas fa-long-arrow-alt-
left'></i>                vorheriger                Monat",
p=kalender&y=$YEAR_PREVIOUS&m=$MONTH_PREVIOUS");
?>
```

*Quellcode 20: Hinzufügen eines Buttons, welcher den anzuzeigenden Zeitrahmen um einen Monat in die Vergangenheit verschiebt, indem er die Kalenderseite mit bestimmten Parametern aufruft*

```
<?php
$CALENDAR_BUTTONS->ADD_BUTTON_CALENDAR_MULTI("mehrtägiges
Ereignis", "?p=kalender&y=$YEAR&m=$MONTH",$CATEGORIES_STR,
$PERMISSIONS_STR,$USERGROUPS_STR);
?>
```

*Quellcode 21: Hinzufügen eines Buttons, welcher eine MessageBox öffnet, in der ein mehrtägiges Ereignis erstellt werden kann*

Die eigentliche Fläche des Kalenders wird durch eine Tabelle realisiert. Die einzelnen Zellen der Tabelle werden durch einen Algorithmus nacheinander mit allen Ereignissen gefüllt, welche innerhalb des Monats liegen, der im Moment geöffnet ist.

Bei der Darstellung ist jedoch zu beachten, dass der erste Tag eines Monats nicht unbedingt ein Montag ist, wodurch wir Platzhalter verwenden mussten, um z.B. die erste Zelle, welche Montag zugeordnet ist, zu überspringen, wenn der 1. Tag im Monat ein Dienstag ist.

```
<?php
$PLACEHOLDERS_COUNT=date("N",strtotime("$YEAR-$MONTH-01")) - 1;
$PLACEHOLDERS;
for($i = 0; $i < $PLACEHOLDERS_COUNT; $i++) {
    $PLACEHOLDERS.="<td class=\"disabled\"></td>";
}
?>
```

*Quellcode 22: Generieren der Anzahl an benötigten Platzhaltern aus dem Rückgabewert der Date-Funktion von PHP, welcher mit dem Parameter "N" und der Zeitangabe des ersten Tages eines Monats numerisch den Wochentag dieses Tages ist und demzufolge der Wert der Funktion - 1 die benötigte Anzahl an Platzhaltern sein muss*

Nach dieser Generierung des allgemeinen Gerüsts des anzuzeigenden Monats wird jeder Tag des Monats auf Ereignisse in der „calender“-Tabelle untersucht. Diese Ereignisse werden dann den jeweiligen Tagen bzw. Tabellenzellen zugeordnet. Bei der Zuordnung wird jedoch zwischen ganztägigen, zeitlich begrenzten eintägigen und mehrtägigen Ereignissen unterschieden. Diese Unterschiede werden durch unterschiedliche Eigenschaften des Fieldsets<sup>17</sup> dargestellt, mit welchem Ereignisse jeweils innerhalb der Tabellenzelle dargestellt und unterteilt werden.

```
<?php
if($EVENTS_SINGLE[$j]["time"]=="NULL"){
$EVENTS_CODE.="<fieldset><legend><p>ganzt&auml;gig</p>";
} else {
$EVENTS_CODE.="<fieldset><legend><p>". $EVENTS_SINGLE[$j]["time"] . "</p>";
}
?>
```

*Quellcode 23: Unterscheidung zwischen ganztägigen Ereignissen und Ereignissen, welche einen bestimmten Zeitrahmen innerhalb eines Tages einnehmen, mit unterschiedlicher Darstellung des Ereignisses durch die Anzeigewerte "ganztägig" oder "<Zeit vom Ereignis>" jeweils innerhalb eines Fieldsets*

Das Hinzufügen von Ereignissen erfolgt innerhalb des Kalenders entweder über die Buttons am Anfang der Seite oder über das Plus, welches innerhalb jedes Tages angezeigt wird. Das Hinzufügen von mehrtägigen Ereignissen oder Kategorien kann jedoch nur über die erstere Methode erfolgen.

---

<sup>17</sup> Fieldset ist ein Element von HTML, welches Informationen unter einer Überschrift sortiert

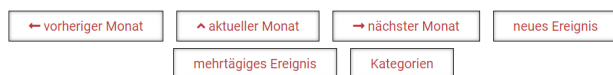


Abbildung 24: Kalendernavigation mit verschiedenen Buttons, welche z.B. das Erstellen von Ereignissen ermöglichen

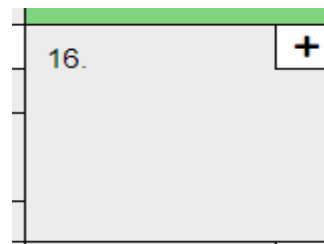


Abbildung 25: Ansicht eines einzelnen Tages im Kalender mit Button zum Hinzufügen von Ereignissen

Das Hinzufügen eines eintägigen Ereignisses z.B. erfolgt über eine MessageBox, in welche sich Eingabeoptionen für den Namen, die Zeit und die zugewiesene Kategorie sowie weitere Konfigurationsmöglichkeiten des Ereignisses befinden.

Ereignis hinzufügen		X
Ereignis:	<input type="text" value="Neues Ereignis"/>	
Datum:	<input type="text" value="tt.mm.jjjj"/>	
Zeit:	<input type="text" value="--:--"/>	
Kategorie:	<input type="text" value="Standard"/>	
Öffentliche Nutzerberechtigung:	<input type="text" value="nichts"/>	
Authentifizierte Nutzerberechtigung:	<input type="text" value="ansehen"/>	
Spezielle Gruppe:	<input type="text" value="Administratoren"/>	
Spezielle Gruppeberechtigung:	<input type="text" value="alles"/>	
		<input type="button" value="Abbrechen"/> <input type="button" value="Bestätigen"/>

Abbildung 26: MessageBox des Kalenders mit Konfigurationsmöglichkeiten des zu erstellenden Ereignisses

Durch das Bestätigen der MessageBox wird der Website zusätzlich zu den eingestellten Konfigurationsvariablen die Variable „func\_add“ übergeben, welche die Routine zum Speichern von Ereignissen startet.

Diese Funktion liest nach einer Rechteabfrage des Users für das Bearbeiten des Kalenders die übergebenen Variablen aus und erstellt einen neuen Eintrag in der Berechtigungs-Tabelle. Die dabei erzeugte Berechtigungs-ID wird anschließend zusammen mit den ausgelesenen Variablen in die „Calendar“-Tabelle eingetragen. Nach dieser Abfolge wird die Seite neu geladen und das neue Ereignis angezeigt.



```

<?php
if(isset($_GET["func_add"])){
if($SPECIAL_PAGE_PERMS == "ALTER" or $SPECIAL_PAGE_PERMS == "EDIT") {
$NEW_DATE=$_GET["func_add"];
$NEW_EVENT=$_GET["event"];
$NEW_TIME=$_GET["time"];
$NEW_CATEGORY=SQL_READ("calendar_categories","id,category","category",
$_GET["category"])[ "id"];
$NEW_PERM_PUBLIC=SQL_READ("permission_levels","internal,display","display",
$_GET["perm_public"])[ "internal"];
$NEW_PERM_AUTH=SQL_READ("permission_levels","internal,display","display",
$_GET["perm_auth"])[ "internal"];
$NEW_PERM_GROUP_ID=SQL_READ("usergroups","id,description","description",
$_GET["group"])[ "id"];
$NEW_PERM_GROUP=SQL_READ("permission_levels","internal,display","display",
$_GET["perm_group"])[ "internal"];
$PERMID=PERMSYS_MAKE_ENTRY($NEW_PERM_PUBLIC,$NEW_PERM_AUTH,
$NEW_PERM_GROUP,$NEW_PERM_GROUP_ID);
SQL_ADD("calendar","date,time,event,category,permissionid","'$NEW_DATE','$NEW_TIME',
'$NEW_EVENT','$NEW_CATEGORY','$PERMID'");
}
header("Location: ?p=kalender&y=$YEAR&m=$MONTH");
}
?>

```

*Quellcode 24: Routine des Kalenders, welche neue Beiträge speichert*

Um Ereignisse zu verändern muss der Stift-/Bearbeitungsbutton des jeweiligen Ereignisses angeklickt werden. Dadurch öffnet sich eine MessageBox, welche Eingabefelder bietet in denen die Konfiguration eines Ereignisses erneuert werden kann.

*Abbildung 27: MessageBox zum Bearbeiten von Ereignissen innerhalb des Kalenders*

Das Bestätigen der MessageBox führt ähnlich wie beim Erstellen von Ereignissen zu einer speziellen Funktion im Kalender. Diese Funktion heißt „func\_change“. Sie überprüft zuerst die Rechte des angemeldeten Nutzers im Bezug auf das Bearbeiten von Ereignissen innerhalb des Kalenders und ändert dann den Eintrag des Ereignisses innerhalb der „calendar“-Tabelle. Nach diesem Ablauf wird die Seite auch hier neu geladen und die Änderungen werden sichtbar.

Bei der Bearbeitung wird auf die Möglichkeit der zeitlichen Veränderung der Ereignisse, auf Grund von unvorteilhaften SQL-Tabellen, verzichtet.

```

<?php
if(isset($_GET["func_change"])){
$CALENDAR_PERM=PERMSYS_GET_PERM_FOR_USER(SQL_READ("calendar","id,per
missionid","id",$_GET["eventid"])[0]["permissionid"]);
if($CALENDAR_PERM == "ALTER" or $CALENDAR_PERM == "EDIT") {
if(isset($_GET["eventid"])){
SQL_CHANGE("calendar","id",$_GET["eventid"],"event",$_GET["func_change"]);
}
}
header("Location: ?p=kalender&y=$YEAR&m=$MONTH");
}

?>

```

*Quellcode 25: Bearbeitungsroutine für Ereignisse innerhalb des Kalenders*

Das Löschen von Ereignissen erfolgt innerhalb der MessageBox zum Bearbeiten von Ereignissen über den „Event Löschen“-Button, welcher die Routine „func\_delete“ startet. Nach einer Überprüfung der erforderlichen Rechte zum Löschen von Ereignissen im Kalender wird zuerst die Permission-ID des Ereignisses ausgelesen, dann das Ereignis über seine ID in der Kalender-Tabelle gelöscht und danach die Konfigurationen der Berechtigungen über die Permission-ID entfernt. Das Entfernen der Berechtigungen verhindert das Ansammeln von Datenmüll innerhalb der Datenbanken und sichert so die Performance der Seite. Auch hier wird anschließend die Seite neu geladen um die Änderungen anzeigen zu lassen.

```

<?php
if(isset($_GET["func_delete"])){
$PID=SQL_READ("calendar","id,permissionid","id",$_GET["func_delete"])[0]["permissionid"];
$CALENDAR_PERM=PERMSYS_GET_PERM_FOR_USER($PID);
if($CALENDAR_PERM == "ALTER") {
SQL_DELETE("calendar","id",$_GET["func_delete"]);
PERMSYS_DELETE($PID);
}
header("Location: ?p=kalender&y=$YEAR&m=$MONTH");
}

?>

```

*Quellcode 26: Routine um Ereignisse des Kalenders zu löschen und deren Berechtigungen zu entfernen*

## 5 Zusammenfassung/Schluss

Zusammenfassend kann man sagen, dass unsere Arbeit den geforderten Merkmalen im Allgemeinen entspricht. Es existiert ein Front-End-Bereich und einen Back-End-Bereich mit jeweils unterschiedlichen Funktionen und der Datenschutz wurde, z.B. durch ein Berechtigungssystem mit Einbindung in die einzelnen Funktionen der Website, realisiert. Die Informationsbereitstellung kann durch ein Seitensystem, einen Kalender, eine Kollegiumsseite und einen Vertretungsplan erfolgen. Mit den Funktionen im Back-End-Bereich kann die Konfiguration der Website verändert werden und auch Menüstrukturen, sowie Benutzerkonten lassen sich verwalten.

Im Vergleich zur aktuellen Website sind viele unserer Umsetzungen vom Umfang der Funktionen den aktuell bestehenden Tools unterlegen. So ist der alte Editor umfangreicher anpassbar und ermöglicht verschiedene Zusatzfunktionen, wie z.B. das Hochstellen von Zeichen. Unsere Website setzt im Gegensatz dazu den Schwerpunkt auf die Grundfunktionen, welche zum Verwenden der Website benötigt werden und bemüht sich dabei um ein möglichst simples Design. Dieser Fakt kann positiv oder negativ ausgelegt werden. Wir wollten mit unserer praktischen Umsetzung aber vor allem die Usability der Website und deren Pflegeleichtigkeit erhöhen und denken durch eine einfache Bedienung der Funktionen und einem verringerten Funktionsumfang dieses Ziel erreicht zu haben.

Ein Vorteil der neuen Website ist z. B. das besser an Mobilgeräte angepasste Design, das Responsive-Design, welches Inhalte im Gegensatz zur alten Website sehr überschaubar darstellen kann. Außerdem werden durch das Speichern jeglicher Informationen innerhalb der Datenbank die Aspekte des Datenschutzes besser umgesetzt und z. B. Kalenderereignisse nicht mehr über den Drittanbieter Jevents verwaltet werden. Auch können Stundenplandaten vollständig intern verarbeitet und angezeigt werden und es ist keine Verlinkung zu Stundenplan24 mehr nötig. Die Nachvollziehbarkeit von Änderungen innerhalb der Webanwendung ist außerdem verbessert wurde, indem ein Logging-System in den Back-End-Bereich eingepflegt wurde, welches verschiedenste Ereignisse protokolliert. Mit der Einführung von Benutzerkonten für jeden Schüler kann weiterhin davon ausgegangen werden, dass interne Daten nicht mehr durch Außenstehende abgerufen werden können und das eine bessere Verteilung von Informationen auf verschiedene Zugriffsebenen erfolgen kann.

In Sachen Performance ist die neue Website jedoch der alten unterlegen. So laden z.B. die Bilder der Banner noch relativ lange. Die verwendeten Banner sind jedoch nur Platzhalter für spätere Bilder und sollen bei Übernahme der Website natürlich erneuert und dann auch sinnvoll komprimiert werden um dieses Problem zu lösen. Weiterhin müssten die Scripte und Ressourcen, welche im Hintergrund geladen werden, erst nach dem Laden der Website abgerufen werden um die initiale Ladezeit enorm zu erhöhen.

In Zukunft kann mit der Website trotzdem zuverlässig gearbeitet werden und selbst wenn Fehler auftreten bzw. Unklarheiten bestehen, denken wir diese durch einen zeitlich begrenzten Support aus der Welt schaffen zu können.

Die Website ist zusätzlich dazu, dass sie im Rahmen des Schulalltags eingesetzt werden kann, eine Grundlage für potentielle weiterführende Leistungen. Der Code ist verständlich kommentiert und möglichst einfach strukturiert worden, um Unbeteiligten ein möglichst großes Verständnis der Funktionsweise unserer Website zu geben. Dieser Fakt ermöglicht eine weiterführende Verwaltung der Website, selbst wenn Änderungen im Code nötig sind.

Mögliche Verbesserungsansätze bzw. weiterführende Ideen sind die bessere Umsetzung der Verarbeitung von mehrtägigen Ereignissen im Kalender oder auch die Entwicklung von Hierarchien der Benutzergruppen mit ihren jeweiligen Rechten. Ein anderer Ansatz wäre außerdem das Optimieren von Ladezeiten mit bereits beschriebenen Methoden oder anderen Ideen.

## 6 Zeitaufwandsnachweis

Datum	Art der Tätigkeit	Zeit
13.12.2018	Erstes Gespräch mit Herr Schreiber, um Thema und Umfang der Arbeit grob zu besprechen	ca. 2h
20.12.2018	Erstellen einer Umfrage über gewünschte Funktionen auf der aktuellen Website und Auswerten der gesammelten Daten („ <a href="https://www.gmg.julius-kuehn.de/umfrage/">https://www.gmg.julius-kuehn.de/umfrage/</a> „,)	ca. 4h
Februar – April 2019	Gespräche mit Lehrern, Schülern und Herrn Gärtner über gewünschte Funktionen und das Design der Arbeit, sowie das Beschaffen von Testdateien des Vertretungsplans und Informationen zur aktuellen Website	ca. 10h
05.01.-29.01.2019	Ideensammeln zu zweit und Recherche im Internet nach Inspiration bei anderen Schulwebsites	pro Woche ca. 4h
06.02.2019	Ratschläge und Tipps von Herr Schreiber und Gespräche über Umsetzung	ca. 1h
07.02.-26.02.2019	Erstellen der allgemeinen Struktur des Website und erste Ansätze des Back-End-Bereichs umsetzen	ca. 12h
02.03.2019	Erstes Ausarbeiten von Grundlagen für die Theorie	ca. 3h
13.03.-21.03.2019	Entwicklung und Integration von Datenbankstrukturen in die Funktionen des Back-End-Bereichs	ca. 7h
06.04.-09.04.2019	Bugs fixen und verschiedenes Umstrukturieren von Code, sowie designtechnische Anpassungen des kompletten Back-End-Bereichs	pro Tag ca. 1,5h
27.04.-06.05.2019	Entwicklung des Berechtigungssystems mit Log-In und Anbindung an Benutzeraccounts	ca. 14h
07.05.2019	Sichten der Dokumentation und Aneignen von jQuery	ca. 2h
15.05.-29.05.2019	Ausarbeiten von theoretischen Grundlagen aus Lektüre und der Handreichung zum Datenschutz	ca. 15h
02.06.2019	Übernahme von groben Strukturen und Funktionen aus dem Back-End-Bereich in den Front-End-Bereich	ca. 2h
13.06.-24.06.2019	Entwickeln erster Funktionen um Seiten darzustellen und zwischen diesen zu navigieren	ca. 8h

02.07.2019	Gespräch mit Herr Schreiber über verschiedene Möglichkeiten einzelne Aspekte umzusetzen	ca. 2h
12.07.-14.07.2019	Aneignung von Methoden des XSS um Erfahrungen in die Website einfließen zu lassen	ca. 4h
24.07.-27.07.2019	komplette Neustrukturierung des Front-End-Bereichs mit Anbinden bereits fertiger Funktionen	ca. 10h
03.08.-05.08.2019	Einbindung von Editor und Seitenübersicht in den Front-End-Bereich und Anpassen von diesem an die gewünschte Darstellung und Funktionen	ca. 5h
14.08.-19.08.2019	Ergänzen des theoretischen Teils der Leistung mit neuen Aspekten zur Umsetzung	ca. 8h
04.10.-01.10.2019	Entwicklung von ersten Versionen des Kalenders und des Klausurplans mit Anbindung an die jeweiligen Datenbanken	pro Tag ca. 3-4h
12.11. - 14.11. 2019	Anbindung aller Funktionen an das Berechtigungssystem	ca. 5h
27.11. 2019	Erstellen des Stundenplan-Moduls und verschiedener Spezialseiten, wie z.B. die Profilseite	ca. 7h
20.12.-23.12.2019	Einarbeitung von Veränderungen der Website in die Theorie	ca. 4h
01.01.-03.01.2020	Einbinden von Code-Ausschnitten in die Theorie mit ergänzenden Erklärungen und Visualisierung verschiedener Beschreibungen mit Bildern	ca. 6h
16.01.2020	Gespräch mit Herr Schreiber: verschiedene Fehler beheben und Server umkonfigurieren lassen	ca. 2h
18.01. -20.01. 2020	Umgestaltung von Funktionen für besser Usability und Hinzufügen von vielen kleineren Details	ca. 6h
29.01. - 14.02. 2020	Vervollständigen der Theorie mit Einbindung aller praktischen Aspekte	ca. 25h

## 7 Quellen

### 7.1 Literaturverzeichnis

Nr.	Titel	Verlag	Autor	ISBN (10)
[01]	Content Management Systeme: Begriffsstruktur und Praxisbeispiel	Springer Gabler	Stefan Spörrer	3937404740
[02]	Einführung in SQL 2. Auflage	O'Reilly	Alan Beaulieu	3897219379
[03]	MySQL Einführung, Programmierung, Referenz 2. Auflage	Addison-Wesley	Michael Kofler	3827320461
[04]	JavaScript The Definitive Guide Fifth Edition	O'Reilly	David Flanagan	0596101996
[05]	Learning jQuery 1.3 (English Edition)	Packt Publishing	Jonathan Chaffer	1847196705
[06]	JavaScript & Ajax Das umfassende Handbuch 2. Auflage	Galileo Computing	Christian Wenz	3898428591
[07]	HTML&XHTML Das umfassende Handbuch	O'Reilly	Chuck Musciano	3897214946
[08]	Einstieg in PHP 5.5 und MySQL 5.6	Galileo Computing	Thomas Theis	3836224895
[09]	CSS Das umfassende Handbuch	O'Reilly	Eric A. Meyer	3897214938
[10]	Programmieren mit PHP	O'Reilly	Rasmus Lerdorf	3897214733
[11]	Einführung in PERL	O'Reilly	Randal L. Schwartz	386899145X
[12]	Apache 2.4 Das umfassende Handbuch	Galileo Computing	Sascha Kersken	3836217775
[13]	Handreichung – Datenschutz an Schulen Version 1.0 Stand: 30.07.2018	O'Reilly	Ministerium für Bildung Sachsen-Anhalt	-

Nummer	URL	Zugriff
[14]	<a href="https://www.bundesverfassungsgericht.de/SharedDocs/Entscheidungen/DE/1983/12/rs19831215_1bvr020983.html">https://www.bundesverfassungsgericht.de/SharedDocs/Entscheidungen/DE/1983/12/rs19831215_1bvr020983.html</a>	13.02.2020
[15]	<a href="https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32001R0045">https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32001R0045</a>	13.02.2020
[16]	<a href="http://www.mathematik.uni-ulm.de/sai/ws01/portalsem/rene/internetportale/node5.html">http://www.mathematik.uni-ulm.de/sai/ws01/portalsem/rene/internetportale/node5.html</a>	27.11.2019
[17]	<a href="https://www.php.net/manual/de/intro-what-is.php">https://www.php.net/manual/de/intro-what-is.php</a>	13.02.2020
[18]	<a href="https://w3techs.com/technologies/overview/programming_language/all">https://w3techs.com/technologies/overview/programming_language/all</a>	13.02.2020
[19]	<a href="https://stackoverflow.com/questions/9948008/what-is-sapi-and-when-would-you-use-it">https://stackoverflow.com/questions/9948008/what-is-sapi-and-when-would-you-use-it</a>	13.02.2020
[20]	<a href="https://www.w3schools.com/cssref/css3_pr_transform.asp">https://www.w3schools.com/cssref/css3_pr_transform.asp</a>	13.02.2020
[21]	<a href="https://www.w3schools.com/cssref/css_selectors.asp">https://www.w3schools.com/cssref/css_selectors.asp</a>	13.02.2020
[22]	<a href="https://de.wikipedia.org/wiki/Cascading_Style_Sheets">https://de.wikipedia.org/wiki/Cascading_Style_Sheets</a>	13.02.2020
[23]	<a href="https://www.bundesverfassungsgericht.de/SharedDocs/Entscheidungen/DE/1983/12/rs19831215_1bvr020983.html">https://www.bundesverfassungsgericht.de/SharedDocs/Entscheidungen/DE/1983/12/rs19831215_1bvr020983.html</a>	13.02.2020
[24]	<a href="https://bdsg-externer-datenschutzbeauftragter.de/datenschutz/was-ist-datenschutz/">https://bdsg-externer-datenschutzbeauftragter.de/datenschutz/was-ist-datenschutz/</a>	13.02.2020
[25]	<a href="https://www.datenschutz-grundverordnung.eu/grundverordnung/art-1-ds-gvo/">https://www.datenschutz-grundverordnung.eu/grundverordnung/art-1-ds-gvo/</a>	13.02.2020
[26]	<a href="https://html.spec.whatwg.org/multipage/parsing.html">https://html.spec.whatwg.org/multipage/parsing.html</a>	13.02.2020
[27]	<a href="https://www.html5rocks.com/de/tutorials/internals/howbrowserswork/#HTML_Parser">https://www.html5rocks.com/de/tutorials/internals/howbrowserswork/#HTML_Parser</a>	13.02.2020
[28]	<a href="https://www.gmg.julius-kuehn.de/">https://www.gmg.julius-kuehn.de/</a>	13.02.2020

verwendete Software	Link
Adminer - Datenbanksoftware	<a href="https://www.adminer.org/de/">https://www.adminer.org/de/</a>
Dreamweaver - HTML Suite	<a href="https://www.adobe.com/de/products/dreamweaver.html">https://www.adobe.com/de/products/dreamweaver.html</a>
FTP Software - Filezilla	<a href="https://filezilla-project.org/">https://filezilla-project.org/</a>

## 7.2 Abbildungsverzeichnis

Alle Abbildungen, Darstellungen und jeglicher Quellcode sind, wenn nicht anders bezeichnet, selbst erstellt. Bilder/Screenshots sind dabei von der alten und der neuen Schulwebsite entstanden.

Abbildung 01: Darstellung eines Beitrages der "News"-Seite, welche sich in der Kategorie "Allgemeines" befindet.....	6
Abbildung 02: Darstellung des Kalenders von JEvents.....	6
Abbildung 03: Backendbereich mit einzelnen Komponenten.....	7
Abbildung 04: Ansicht der Kategorienverwaltung.....	8
Abbildung 05: Ansicht der Seitenübersicht mit Verlinkung zum Editor über den Namen der Seiten, bzw. über die Erstellung einer neuen Seite.....	8
Abbildung 06: Ansicht einer Seite im Editor.....	9
Abbildung 07: Ansicht der einzelnen Menüs mit unterschiedlichen Bearbeitungsmöglichkeiten. .	9
Abbildung 08: Medienverwaltung von Joomla.....	10
Abbildung 09: Medienverwaltung von "JCE".....	10
Abbildung 10: Errormeldung nach Interaktion mit gesperrtem Beitrag.....	10
Abbildung 11: Joomla Modul "Kuenta Forum" in der Ansicht des Frontend.....	10
Abbildung 12: Login erforderlich bei Aufruf einer Seite, welche höhere Rechte erfordert ("https://www.gmg.julius-kuehn.de/index.php/beitrag-erstellen1").....	11
Abbildung 13: fehlgeschlagener Login bei Eingabe eines falschen Passworts.....	11
Abbildung 14: Performance Test der aktuellen Schulwebsite mit der Testsoftware Audits.....	11
Abbildung 15: Performance Test der neuen Schulwebsite mit der Testsoftware Audits in der Entwicklungsphase.....	11
Abbildung 16: Eintrag der Beitragsübersicht z.B. mit Angabe des Autors, Veränderungen werden jedoch nicht dokumentiert.....	11
Abbildung 17: Link zum Vertretungsplan auf der Schulwebsite.....	12
Abbildung 18: Authentifizierung durch Drittanbieter Indivare um auf Vertretungsplan der Schule zuzugreifen.....	12
Abbildung 19: Ansicht des Vertretungsplans bei Eingabe korrekter Benutzerdaten.....	12
Abbildung 20: Direktlink zum Vertretungsplan ermöglicht das unbemerkte Umgehen der Benutzerdatenüberprüfung.....	12
Abbildung 21: Ansicht des Vertretungsplans bei Eingabe des Direktlinks.....	12
Quellcode 01: Hinzufügen einer Seite zur "pages"-Tabelle mit vorheriger Formatierung des Bezeichners.....	19
Quellcode 02: Funktionsaufruf um neuen Eintrag in der Berechtigungs-Tabelle zu erstellen und die neue Berechtigungs-ID zu erhalten.....	19
Quellcode 03: Funktion "PERMSYS_MAKE_ENTRY()" legt neue Berechtigungen innerhalb der „permissions“-Tabelle an.....	19
Abbildung 22: Konfiguration der Berechtigungseinstellungen einer Seite im Editor.....	20
Quellcode 04: Auslesen der jeweiligen ID der im Editor ausgewählten Eigenschaften um diese später als Verweis zu anderen Tabellen verwenden zu können.....	20
Quellcode 05: Verändern von Berechtigungen eines Eintrages der "permissions"-Tabelle mit Hilfe der vorher ausgelesenen ID's der eingegebenen Optionen.....	21
Quellcode 06: Fallunterscheidung der Berechtigungen durch das zurückgegebene Berechtigungslevel der Funktion "PERMSYS_GET_PERM_FOR_USER()".....	22



Quellcode 07: Funktion der Website, welche das maximale Berechtigungslevel einer Nutzers im Zusammenhang mit einer Seite zurückgibt.....	23
Quellcode 08: Funktion um Konfigurationsdaten zu überprüfen.....	24
Abbildung 23: Buttons des Vertretungsplans.....	24
Quellcode 09: Erstellen von groben Strukturen des Vertretungsplans.....	24
Quellcode 10: Laden der Vertretungsdaten über eine Ajax-Anfrage und Weiterleitung zur "applySchedule"-Funktion.....	25
Quellcode 11: schematisches Auslesen der XML-Datei und Einfügen in Variablen.....	26
Quellcode 12: Erstellen einer Liste mit für den Schüler relevanten bzw. allen Informationen....	26
Quellcode 13: Ausgabe der Daten des XML-Dokuments als Tabellenzeilen.....	27
Quellcode 14: Ausgabe aller möglichen Klassen innerhalb eines "select"-Elements.....	27
Quellcode 15: Ausgabe aller möglichen Kurse innerhalb eines benutzerdefinierten "checkbox"-Elements.....	28
Quellcode 16: Funktion um gewählte Kurs zu strukturieren.....	29
Quellcode 17: Funktion um bereits Ausgewählte Kurse hervorzuheben.....	29
Quellcode 18: Standardwert der Zeitzone verändern mit Hilfe einer Funktion von PHP.....	29
Quellcode 19: Festlegen des anzuzeigenden Jahres entweder über den übermittelten Parameter oder über das aktuelle Jahr.....	30
Quellcode 20: Hinzufügen eines Buttons, welcher den anzuzeigenden Zeitrahmen um einen Monat in die Vergangenheit verschiebt, indem er die Kalenderseite mit bestimmten Parametern aufruft.....	30
Quellcode 21: Hinzufügen eines Buttons, welcher eine MessageBox öffnet, in der ein mehrtägigen Ereignis erstellt werden kann.....	30
Quellcode 22: Generieren der Anzahl an benötigten Platzhaltern aus dem Rückgabewert der Date-Funktion von PHP, welcher mit dem Parameter "N" und der Zeitangabe des ersten Tages eines Monats numerisch den Wochentag dieses Tages ist und demzufolge der Wert der Funktion - 1 die benötigte Anzahl an Platzhaltern sein muss.....	31
Quellcode 23: Unterscheidung zwischen ganztägigen Ereignissen und Ereignissen, welche einen bestimmten Zeitrahmen innerhalb eines Tages einnehmen, mit unterschiedlicher Darstellung des Ereignisses durch die Anzeigewerte "ganztägig" oder "<Zeit vom Ereignis>" jeweils innerhalb eines Fieldsets.....	31
Abbildung 24: Kalendernavigation mit verschiedenen Buttons, welche z.B. das Erstellen von Ereignissen ermöglichen.....	32
Abbildung 25: Ansicht eines einzelnen Tages im Kalender mit Button zum Hinzufügen von Ereignissen.....	32
Abbildung 26: MessageBox des Kalenders mit Konfigurationsmöglichkeiten des zu erstellenden Ereignisses.....	32
Quellcode 24: Routine des Kalenders, welche neue Beiträge speichert.....	33
Abbildung 27: MessageBox zum Bearbeiten von Ereignissen innerhalb des Kalenders.....	33
Quellcode 25: Bearbeitungsroutine für Ereignisse innerhalb des Kalenders.....	34
Quellcode 26: Routine um Ereignisse des Kalenders zu löschen und deren Berechtigungen zu entfernen.....	34
Abbildung 28: schematische Darstellung der Funktionsweise von CMS.....	47
Abbildung 29: CMS – selektive Auswahl Drop-Down-Auswahl.....	47
Quellcode 27: CMS - selektive Auswahl Codestruktur.....	48
Abbildung 30: Seitenübersicht im Back-End-Bereich.....	48

Abbildung 31: Seitenübersicht innerhalb der MySQL Tabelle.....	48
Abbildung 32: Beitragsinhalt 'Startseite' Editor-Ansicht.....	49
Abbildung 33: Beitragsinhalt 'Startseite' Frontend-Ansicht.....	49
Quellcode 28: allgemeine Struktur von HTML-Dokumenten.....	51
Quellcode 29: Code Beispiel von "https://www.html5rocks.com/de/tutorials/internals/howbrowserswork/ #HTML_Parser" zuletzt abgerufen am 11.02.2020.....	52
Quellcode 30: Ausgabe eines Textes durch PHP innerhalb eines HTML-Dokuments.....	54
Abbildung 34: Ausgabe der "echo"-Funktion kombiniert mit der Ausgabe von HTML.....	54
Abbildung 35: CSS Syntax.....	56
Abbildung 36: Beispiel für verschiedene CSS Syntax.....	57
Abbildung 37: schematische Darstellung der Browserunterstützung einer JavaScript-Funktion.	58
Quellcode 31: in HTML Kontext integrierter JavaScript-Code.....	59
Quellcode 32: beispielhafter Aufbau eines HTML-Dokuments mit externem JavaScript-Code.	60
Quellcode 33: externer JavaScript-Code, welcher im HTML-Dokument ausgeführt wird.....	60
Quellcode 34: beispielhafter Vergleich von JavaScript mit jQuery.....	62
Abbildung 38: Darstellung von Informationen im JSON-Format.....	63
Quellcode 35: PHP-Datei "datei.php" erstellt Array und gibt dieses zurück.....	64
Quellcode 36: HTML-Datei "index.html" führt AJAX-Anfrage aus und stellt das übergebene Array dar.....	64
Abbildung 39: Ausgabe der SQL-Anweisung "SELECT * FROM `metadata`" in Tabellenform	66
Quellcode 37: beispielhafte Schemaanweisung zum Erstellen einer neuen Tabelle mit zwei verschiedenen Datentypen.....	66
Quellcode 38: beispielhafte Schemaanweisung zum Einfügen von Daten in eine Tabelle.....	66

## **Danksagung**

Großer Dank für die Hilfe bei der technischen, theoretischen und auch organisatorischen Umsetzung unserer Lernleistung gehen an folgende Menschen, welche alle am erfolgreichen Werdegang der Arbeit beteiligt waren:

**Betreuung:** Toni Schreiber

**Fachlehrer:** Frau Bornschein

Auch wollen wir unseren Familien danken, dass sie uns während der langen Zeit der Lernleistung immer den Rücken freigehalten und zu Neuem inspiriert haben.

## Selbständigkeitserklärung

Hiermit erklären wir, dass wir die besondere Lernleistung ohne fremde Hilfe angefertigt und nur die im Quellenverzeichnis angeführten Hilfsmittel verwendet haben.

-----

(Ort, Datum)

-----

-----

(Ort, Datum)

-----

## **A Anhang/ Grundlagen**

### **A.1 Datenschutz allgemein und dessen Relevanz (Quellen: [13], [14], [15], [24] und [25])**

Datenschutz bezeichnet im Allgemeinen die Regelungen zum Schutz von personenbezogener Daten vor Missbrauch, mit welchem weiterhin auch die Regelungen zum Schutz der Privatsphäre einhergehen. Diese sind in Deutschland nach dem Bundesverfassungsgericht Grundrechte jedes Menschen, welche gewährleisten, dass jeder Mensch grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten bestimmt (laut [14]). Durch die Datenschutz-Grundverordnung der Europäischen Union ist der Datenschutz dabei nicht nur in Deutschland, sondern auch in allen anderen Mitgliedsstaaten festgelegt und darf weder verboten, noch eingeschränkt werden.

Im Rahmen des Datenschutzes an deutschen Schulen müssen Anwender über die Nutzung ihrer personenbezogenen Daten in leicht verständlicher Art und Weise unterrichtet werden. Daten sollen nur noch in dem Rahmen erhoben werden, welcher für den Anwendungsfall nötig ist. Ein Beispiel dafür ist die Erstellung eines Benutzerkontos für einen Schüler innerhalb der Schulwebsite. Für die Funktionsweise der Website ist ein Benutzername, ein Passwort, eine E-Mail-Adresse und die Klasse des Schülers notwendig um diesen eindeutig einordnen zu können und ihm z.B. Beiträge für eine bestimmte Nutzergruppe anzuzeigen. Weitere Daten, wie z.B. eine Adresse o.Ä. sollen nach der Datenschutzverordnung nicht mehr erhoben werden. Bei der Übertragung und Speicherung von sensiblen Informationen sollen im Allgemeinen Verschlüsselungen verwendet werden um einen unberechtigten Zugriff auf diese zu verhindern.

Mit zunehmender Datenerfassung im Internet muss der Datenschutz, um seinen Anforderungen gerecht zu werden, auch im technischen Bereich Anwendung finden. Jedoch sind Dateien im Internet theoretisch einfach erreichbar und ein Zugriff auf diese durch Unbefugte kann teilweise ohne fachspezifische Kenntnisse erfolgen. Auf Grund dessen muss dieser Zugriff durch verschiedene Maßnahmen verhindert werden. Diese werden unter dem Begriff Datensicherheit zusammengefasst und stellen die technische Umsetzung der Anforderungen des Datenschutzes dar.

Die Datensicherheit in der Informatik basiert im Grundlegenden auf 3 Aspekten: Vertraulichkeit, Verfügbarkeit, und Integrität. Dabei versteht man unter Vertraulichkeit, dass Daten nur durch dafür berechtigte Personen angezeigt bzw. verändert werden können. Unter Verfügbarkeit versteht man weiterhin das Ziel, ein System so zu gestalten, dass die Ausfallsrate minimiert und die Erreichbarkeitsrate maximiert wird. Der letzte Aspekt ist die Integrität von Daten, welche im Allgemeinen die Nachvollziehbarkeit von Datenänderungen beschreibt und dessen Ziel die Verhinderung von unerkannten Datenänderungen ist.

Aus diesen Aspekt folgt die Notwendigkeit von verschiedenartigen Maßnahmen der Datensicherheit. Die Datensicherheit ist in diesem Kontext der technische Bereich, welcher die Maßnahmen umfasst, welche den Schutz von Daten realisieren. Dabei können z.B. Berechtigungssysteme und die Verschlüsselung von Daten den Zugriff auf diese einschränken und Risikoanalysen einzelner Systeme, mit Angaben über die Wahrscheinlichkeit und die Dauer eines Ausfalls, eine Grundlage für die Analyse der Verfügbarkeit bieten. Anders als der Datenschutz umfasst die Datensicherheit weiterhin nicht nur persönliche, sondern alle Daten.

Im Zuge der Aktualität des Datenschutzes haben wir uns für eine Besondere Lernleistung in diesem Bereich der Informatik entschieden und ihn mit der Entwicklung sowie Gestaltung einer neuen Schulwebsite mit eigenem Content-Management-System(Vgl.: Kap. Content Management System (CMS) (Quelle: [01]), S. 47) gekoppelt. Die Schulwebsite bot sich dabei als praktische Anwendung an, da diese Teil einer öffentlichen Einrichtung ist und der Datenschutz Grundverordnung gerecht werden muss (laut [15]).

Auch ist zu beachten, dass Datenschutz und die damit einhergehende Datensicherheit nicht nur grundlegende Rechte jedes Menschen sind, sondern auch einen Marktfaktor darstellen. So werden z.B. tendenziell mehr Kunden eine sichere Webanwendung einer unsicheren vorziehen. Dadurch ist die Gewährleistung des Datenschutzes eines der wichtigsten Merkmale, welche Webanwendung in der Zeit der Internet- und Telekommunikation erfüllen müssen (laut [13] und [15]).

## **A.2 Datenbank Grundlagen (Quellen: [02], [03] und [08])**

### **A.2.1 Datenbank - Allgemein**

Als Datenbank wird im Allgemeinen jede strukturierte Sicherung von Daten bezeichnet. So ist z.B. ein Wörterbuch oder ein Telefonbuch eine analoge Datenbank für Informationen zu bestimmten Schlüsselbegriffen bzw. zu gelisteten Telefonnummern oder Namen.

Im Kontext der Webprogrammierung versteht man unter einer Datenbank meist ein Datenbanksystem, welches als Speicherort für digitale Informationen zur Verfügung steht, weiterhin jedoch auch entsprechende Funktionen, z.B. über die Verwendung von SQL (siehe A.9 SQL (Quelle: [02]) S. 65) mit sich bringt, um die Bearbeitung, Erstellung und Einbindung von Daten und dazugehörigen Datenstrukturen zu ermöglichen.

### **A.2.2 Datenbank – Klärung wichtiger Begriffe**

#### **A.2.2.1 Datenbank – Datensatz**

Als Datensatz wird jede Zeile einer Tabelle in einer Datenbank bezeichnet. Der Inhalt und das Schema des Datensatzes ist dabei von der Struktur der Tabelle abhängig.

#### **A.2.2.2 Datenbank – Datenfeld**

Ein Datenfeld bezeichnet einen bestimmten Punkt eines Datensatzes. Dabei kann er nur Daten eines Typs speichern bzw. wird durch andere Vorgaben, wie z.B. die Länge von Zeichenketten, eingegrenzt.

#### **A.2.2.3 Datenbank – Query/Abfrage**

Innerhalb des Kontextes einer Datenbank wird als Query/Abfrage eine Auswahl bestimmter Daten bezeichnet, deren Ergebnis eine temporär generierte Tabelle ist. Diese Auswahl erfolgt meist auf Grundlage von in der Tabelle gespeicherten Merkmalen/Werten.

#### **A.2.2.4 Datenbank – Redundanz**

In einer Datenbank versteht man unter dem Begriff Redundanz die mehrfache Existenz von gleicher Daten. Sie ist in Datenbanken möglichst zu vermeiden, um einen effizienten Zugriff auf die darin gespeicherten Daten zu ermöglichen.

#### **A.2.2.5 Datenbank – Primärschlüssel**

Unter einem Primärschlüssel versteht man eine bestimmte Gruppe von Werten, welche dazu dienen, verschiedene Daten innerhalb einer Datenbank eindeutig zu identifizieren. Ein Beispiel ist die fortlaufende einzigartige Nummerierung von Datensätzen, welche häufig als ID bezeichnet wird und einen Datensatz mit einer Nummer eindeutig identifizieren kann.

## A.3 Content Management System (CMS) (Quelle: [01])

### A.3.1 CMS - Allgemein

Ein Content Management System ist ein wichtiger Teil dynamischer Webanwendungen, welche das Pflegen und das Verwalten von Daten in diesen erleichtern soll. Mit ihnen können z.B. Redundanzen verringert werden, indem für bestimmte Daten nur ein Speicherort verwendet wird.

Ein solches System kann dabei fast alle Anforderungen im Zusammenhang mit Datenverarbeitung, also Speicherung, Steuerung, Kontrolle von Daten, sowie die zur Bereitstellung und Verwaltung von Informationen auf der Website erforderlichen Maßnahmen mit minimalem Aufwand durch den Benutzer ausführen. Dieser steuert das System innerhalb einer visuellen Umgebung, welche nur wenig Kenntnisse in Programmiersprachen erfordert und vereinfacht damit den Arbeitsprozess, welcher mit der Pflege einer Website verbunden ist, um ein Vielfaches.

### A.3.2 CMS - Funktionsweise

Die Funktionsweise eines CMS gliedert sich in die Speicherung, die Bearbeitung und die Bereitstellung von Daten. Bei der Speicherung werden Informationen durch den Anwender z.B. in einem Web-Editor eingegeben und anschließend über bestimmte Funktionen des Content Management Systems innerhalb einer Datenbank systematisch gespeichert. Dabei können verschiedene Spezifikationen, wie z.B. Kategorien einzelner Datensätze, eingestellt und mit den Informationen verknüpft werden.

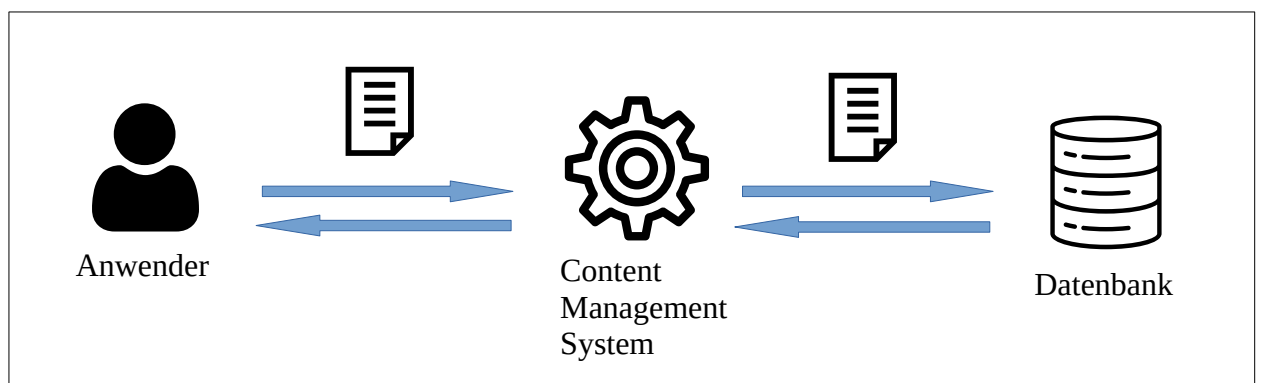


Abbildung 28: schematische Darstellung der Funktionsweise von CMS

Das Bearbeiten von Daten innerhalb eines CMS erfolgt, indem das System die bereits gespeicherten Informationen aus der Datenbank lädt und diese z.B. wieder an einen Web-Editor übergibt, in welchem sich die Daten verändern lassen. Weiterhin kann ein CMS, z.B. durch eine selektive Auswahl, die mit den Datensätzen verknüpften Spezifikationen verändern oder einen Datensatz vollständig aus der Datenbank löschen.

Kategorie:	<div>Standard ▾</div>
Freigegeben:	<div>Ja ▾</div>
Kategorie:	<div>Standard ▾</div>
Freigegeben:	<div>Standard Debugging Test Kategorie</div>

Abbildung 29: CMS – selektive Auswahl Drop-Down-Auswahl

```

▼ <select name="page_category"> == $0
    <option selected>Standard</option>
    <option>Debugging</option>
    <option>Test Kategorie</option>
</select>

```

Quellcode 27: CMS - selektive Auswahl Codestruktur

Die Bereitstellung von Daten kann in einem CMS z.B. durch das Anzeigen in einem Beitrag, also in einer Unterseite der Webanwendung, erfolgen. Datensätze können dabei unterschiedlichste Verwendungen haben. So können nicht nur Sätze, Zahlen, o.Ä., sondern auch Spezifikationen, welche die Darstellung beschreiben, enthalten, über das CMS bereitgestellt werden.

Seitenübersicht						
In diesem Bereich können die Seiten der Website verwaltet und bearbeitet werden.						
<input type="button" value="Seite hinzufügen"/> <input type="button" value="Seite suchen"/>						
ID	Seitenname (alias)	Kategorie	Freigegeben	Autor	Aufrufe	
7	<a href="#">Abmelden</a> (logout_submit)	Spezialseite		admin_gmg	7	
5	<a href="#">Anmelden</a> (login)	Spezialseite		admin_gmg	53	
12	<a href="#">Datenschutz</a> (datenschutz)	Standard	✓	admin_gmg	21	✗
3	<a href="#">Klausurplan</a> (klausurplan)	Spezialseite		admin_gmg	15	
4	<a href="#">Kollegium</a> (kollegium)	Spezialseite		admin_gmg	13	
6	<a href="#">Login-Process</a> (login_submit)	Spezialseite		admin_gmg	22	
8	<a href="#">Profilseite</a> (profil)	Spezialseite		admin_gmg	42	
10	<a href="#">Seitenübersicht</a> (uebersicht)	Spezialseite		admin_gmg	37	

Abbildung 30: Seitenübersicht im Back-End-Bereich

<input type="checkbox"/> Ändern	id	title	page	category	source	active	views	permissionid
<input type="checkbox"/> bearbeiten	1	Seiteneditor	edit	2		1	62	1
<input type="checkbox"/> bearbeiten	2	Terminkalender	kalender	2		1	173	2
<input type="checkbox"/> bearbeiten	3	Klausurplan	klausurplan	2		1	15	3
<input type="checkbox"/> bearbeiten	4	Kollegium	kollegium	2		1	13	4
<input type="checkbox"/> bearbeiten	5	Anmelden	login	2		1	53	5

Abbildung 31: Seitenübersicht innerhalb der MySQL Tabelle



In der Beitragsübersicht wird z.B. die Information, ob eine Seite freigegeben ist, als Haken oder Kreuz angezeigt. Die Informationsgrundlage liegt dabei innerhalb der MySQL-Tabelle im Punkt „active“, welcher mit der Zahl 1 aktiv und mit der Zahl 0 inaktiv darstellt. Diese Spezifikation wird also durch einen Datensatz gespeichert.

In der Tabelle existiert außerdem ein Datenfeld für den Inhalt der Beiträge. Dieser Inhalt kann dabei innerhalb des Frontends und im Editor angezeigt werden. Als Beispiel dient hier der Beitrag mit dem Titel „Startseite“.



Abbildung 32: Beitragsinhalt 'Startseite' Editor-Ansicht

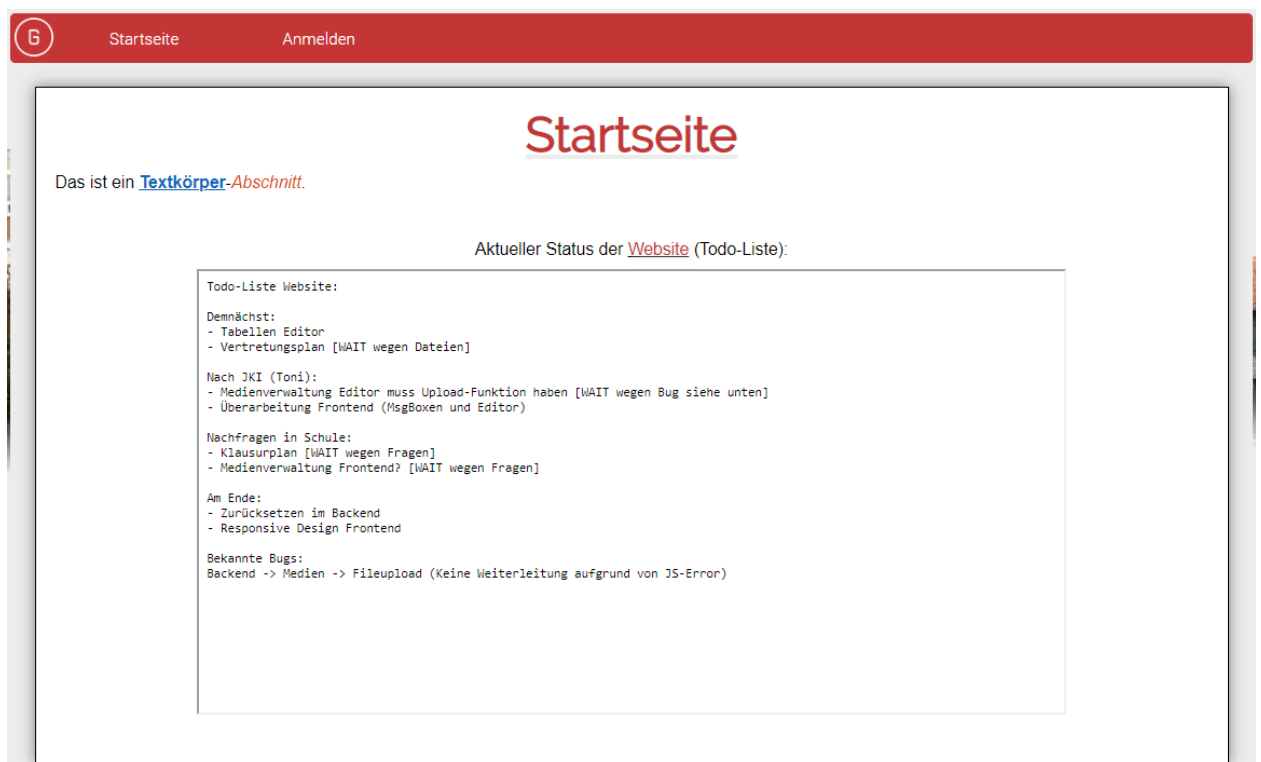


Abbildung 33: Beitragsinhalt 'Startseite' Frontend-Ansicht

### **A.3.3 CMS - Aufbau**

Der Aufbau eines Content Management Systems unterscheidet sich in der Regel von Webanwendung zu Webanwendung, weshalb der Aufbau im Nachfolgenden am Beispiel der Schulwebsite beschrieben wird.

Grundlage unseres CMS ist ein öffentlicher Bereich, in welchem Informationen, wie z.B. einzelne Beiträge dargestellt werden. Diese Beiträge können im Editor bearbeitet werden, welcher auch das Erstellen von neuen Beiträgen, sowie die parallele Speicherung der Daten innerhalb einer Datenbank ermöglicht.

Andere Dateien werden vom CMS der Webanwendung in einer Medienansicht dargestellt. Dort können Dateien gelöscht oder auch hochgeladen werden.

Im Backend-Bereich können z.B. Rechte vergeben oder auch die Meta-Daten-Einstellungen<sup>18</sup> verändert werden. Die jeweiligen Daten und Konfigurationen sind, ähnlich wie Beiträge, innerhalb eines Datenbanksystems gespeichert und werden in diesem verwaltet.

### **A.3.4 CMS – Anwendungsgebiete/Beispiele**

CMS werden heutzutage in den meisten Webanwendungen genutzt. Dabei sind kostenlose Versionen wie z.B. WordPress oder auch Joomla meist die attraktivste Wahl für viele Nutzer. Kostenpflichtige Systeme, mit deutlich mehr spezialisierten Funktionen, wie z.B. Shopify, ein Online-Shop-Anbieter, erfreuen sich jedoch auch steigender Nachfrage. CMS werden also nicht nur zur reinen Informationsbereitstellung, sondern z.B. auch als moderne Shopsysteme genutzt. Weitere Anwendungsgebiete von CMS sind jedoch auch Foren- oder Chatsysteme, deren Informationsbereitstellung durch die automatische Verwaltung überhaupt erst möglich gemacht wird.

## **A.4 HTML („Hypertext Markup Language“) (Quellen: [07], [26] und [27])**

### **A.4.1 HTML – Allgemein**

HTML ist eine standardisierte „markup language“ oder auch „Auszeichnungssprache“, welche die Grundlage des Internets bildet. Durch die einfache und fehlertolerante Syntax der Sprache erfreut sie sich großer Beliebtheit und lässt auch Anfänger z.B. eigene Webauftritte realisieren.

Im Vergleich zu anderen Sprache kann HTML nicht als Programmiersprache bezeichnet werden, da ein HTML-Dokument nur die Art und Weise der Wiedergabe von Bildern, Text und Videos definiert.

In seinem Kern basiert HTML auf der SGML („Standard Generalized Markup Language“), und übernimmt von dieser viele Funktionen, gestaltet jedoch deren Einbindung viel simpler bzw. verzichtet auf komplexe Funktionen.

Meistens wird HTML im Web-Kontext mit CSS (vgl. Kap.: A.6 CSS („Cascading Style Sheets“) (Quellen: [09], [20], [21] und [22]) S. 55), JavaScript (vgl. Kap.: 57 Javascript (JS) (Quellen: [04], [05] und [06]) S. 57) und teilweise auch PHP (vgl. Kap.: A.5 PHP („Hypertext Preprocessor“ bzw. „Personal Home Page Tools“) (Quellen: [08], [10] und [17]) S. 53) erweitert und ist im allgemeinen nur für die Struktur von Webdokumenten zuständig.

Die Weiterentwicklung von HTML ist die „Extensible Hypertext Markup Language“, kurz XHTML, welche z.B. keine Gestaltungselemente mehr enthält und diesen Aspekt direkt an CSS-Dateien auslagert.

---

<sup>18</sup> Meta-Daten-Einstellungen umfassen jegliche Konfiguration der Meta-Tags in HTML, welche z.B. für die optimale Einordnung in die Suchergebnisse von Google verantwortlich sind

## A.4.2 HTML – Funktionsweise

HTML basiert auf der Verwendung von sogenannten „Tags“. Diese sind eine Art Indikator für den Browser, welche dadurch einzelne Inhalte und Teile von anderen Trennen bzw. zusammenfassen. Einige Beispiele dieser Tags sind z.B. „<img>“, welcher dem Browser die Implementierung eines Bildes ankündigt oder auch „<head>“, welcher den Kopf-Abschnitt eines HTML-Dokumentes einleitet. Um beispielsweise den Kopfbereich wieder zu „schließen“, also das Ende dieses Bereichs anzuzeigen, wird ein sogenannter „Closing-Tag“ des einleitenden Tags, in unserem Fall also „</head>“ verwendet. Die meisten Tags in HTML müssen wieder geschlossen werden, da innerhalb der definierten Bereiche oder Elemente andere Elemente eingebunden werden können. Anders ist das nur bei Tags, welche diese Funktionalität nicht besitzen. So benötigt der „Absatz- / <br>-Tag“ kein Closing-Tag, da innerhalb eines Absatzes logischerweise kein anderes Element/ kein anderer Tag eingebunden werden kann. Das Verhalten dieser Tags und die Syntax von HTML sind dabei durch das W3C („World Wide Web Consortium“) vorgegeben und dadurch in den Browser gleich bzw. sehr ähnlich.

Die grundlegende Struktur gliedert sich dabei in den Kopf-Abschnitt und den Hauptteil, den Body-Abschnitt. Innerhalb von HTML gibt es dabei jedoch nur einzelne Elemente, welche nur an bestimmten Stellen stehen dürfen (siehe PHP - Funktionsweise, JS – Funktionsweise, CSS - Funktionsweise).

```
<!DOCTYPE html>

<head>
  <meta charset="utf-8" />
  <title>Titel der Website </title>
  <script src="script.js"></script>
</head>

<body>
  <p>Test Paragraph</p>
  <br>
  <div>
    <a src="website.de"></a>
  </div>
</body>

</html>
```

Quellcode 28: allgemeine Struktur von HTML-Dokumenten

<code>&lt;html&gt; &lt;/html&gt;</code>	Öffnungs-Tag von HTML-Dokumenten
<code>&lt;head&gt; &lt;/head&gt;</code>	Header-Bereich von HTML-Dokumenten
<code>&lt;body&gt;&lt;/body&gt;</code>	Hauptteil von HTML-Dokumenten
<code>&lt;div&gt;&lt;/div&gt;</code>	Bereichs-Tag zur Abgrenzung von einzelnen Bereichen
<code>&lt;br&gt;</code>	Tag um Absatz zu erzeugen (benötigt keinen Closing-Tag)
<code>&lt;title&gt;&lt;/title&gt;</code>	Titel der Website

Im Beispiel erfolgt zuerst die Deklaration eines HTML-Dokumentes, in welchem der restliche Code notiert wird. Der **HTML-Tag** wird demzufolge auch erst am Ende des Dokumentes beendet.

Der **Head-Tag** steht vor dem **Body-Tag** und kann z.B. Angaben zu Meta-Daten (Sprache, Zeichensatz, Indexierungsoptionen, ...), den Titel der Website und auch verschiedene Verweise z.B. zu ausgelagerten JavaScript-Dateien beinhalten. Mit dem „</head>-Tag“ wird dieser Bereich abgeschlossen und es folgt der Body-Bereich, also der Hauptteil des Dokumentes.

Im Body können verschiedenartige Tags stehen. Besonders ist in unserem Fall der „<div>-Tag“, welcher einen abgekapselten Bereich innerhalb eines anderen Bereichs bzw. innerhalb eines andern „<div>-Bereichs“ definiert. Im Beispiel beinhaltet dieser Bereich einen Link.

Im Einzelnen gibt es weiterhin einzelne Tags die spezielle Bereiche definieren, welche innerhalb von HTML eine bestimmte Position besitzen. Darunter zählt z.B. der Footer-Tag, welcher jeweils am Ende eines Bereiches stehen kann.

Die Elemente sind weiterhin in ihrer Anordnung nicht eingeschränkt und auch im Sinne des Syntax werden z.B. unterschiedliche Schreibweise der Tags unterstützt und damit eine relativ große Freiheit bei der Arbeit mit HTML ermöglicht. Diese Fehlertoleranz wird z.B. durch verschiedene Abfragen erzielt. Im Beispiel wird dabei ein </br> als <br> interpretiert. Die Fehlerkorrektur wird durch eine **if-/Wenn-Abfrage** realisiert. Durch zwei weitere Abfragen wird nun die **Existenz von einem geschlossenen <br>-Tag** und die **korrekte Konfiguration des Browsers** für die Korrektur in einen sogenannten Wahrheitswert umgewandelt. Wenn die Abfrage wahr ist, also ein geschlossenen <br>-Tag vorliegt, wird die Routine innerhalb der geschweiften Klammern ausgeführt, ansonsten wird der Code innerhalb der Abfrage außer Acht gelassen. Dabei wird hier bei einem wahren Wert zuerst eine **interne Errormeldung generiert** und der Tag einfach als „Beginning-Tag“, also geöffneter Tag (<br>), **interpretiert**.

```
If (t->isCloseTag(brTag) && m_document->inCompatMode()) {
    reportError(MalformedBRError);
    t->beginTag = true;
}
```

*Quellcode 29: Code Beispiel von  
["https://www.html5rocks.com/de/tutorials/internals/howbrowserswork/#HTML\\_Parser"](https://www.html5rocks.com/de/tutorials/internals/howbrowserswork/#HTML_Parser) zuletzt abgerufen am 11.02.2020*

### **A.4.3 HTML – Aufbau**

HTML basiert im Grundlegenden auf einem Textdokument, welches verschiedene Tags beinhaltet. Der Webbrowser des Anwenders wandelt dieses Dokument mit Hilfe seines eingebundenen HTML-Parsers in ein Webdokument um, welches dem Anwender im Browser angezeigt wird. Die Erklärung des Parsens wird durch seine enorme Komplexität und sein vielfach benötigtes Fachwissen aus der Betrachtung im Kontext der Lernleistung stark verkürzt. Das Parsen erfolgt im groben nach dieser Reihenfolge: HTML-Code parsen, CSS-Code parsen, Skripte parsen, ausgelagerte Dateien parsen, Rendering-Reihenfolge konstruieren, Webdokument rendern.

Nachdem das Webdokument erstellt wurde, wird dieses durch die „Document Object Model“ Schnittstelle, kurz DOM, des Browsers mit anderen Elementen, wie z.B. JavaScript, verbunden und es entsteht ein interaktives Dokument, welches z.B. eine Webanwendung darstellen kann.

Die Abarbeitung der Tags erfolgt dabei nach der Baumstruktur von DOM, welche nach einer vorgegebenen Reihenfolge die Haupttags (<head> , <body> , ...) abarbeitet und dabei in der aus dem HTML-Dokument hervorgehenden Reihenfolge die Tags innerhalb der Haupttags mit ihren teilweise wiederum eingeschlossenen Tags berücksichtigt und zu einem Browser-Dokument umwandelt.

### **A.4.4 HTML – Anwendungsgebiete**

HTML wird fast ausschließlich in der Webentwicklung verwendet, da die standardisierten Definitionen auf diesen Anwendungszweck zugeschnitten sind und die Interpreter für diese Aufgabe optimiert bzw. entwickelt wurden. Weitere Anwendungen sind ansonsten nur im Kontext einer Browserumgebung innerhalb einer anderen Software z.B. bei der Darstellung von Hilfetexten o.Ä. zu finden.

## **A.5 PHP („Hypertext Preprocessor“ bzw. „Personal Home Page Tools“) (Quellen: [08], [10] und [17])**

### **A.5.1 PHP - Allgemein**

PHP oder auch „Hypertext Preprocessor“ ist eine serverseitige Skriptsprache, welche in einer Vielzahl von aktuellen Websites Anwendung findet. Serverseitig bedeutet dabei, dass der Code nicht auf dem lokalen System des jeweiligen Anwenders, sondern auf dem Server ausgeführt wird. Der Client empfängt daraufhin nur noch die Ausgabe des PHP-Codes und zeigt diese im Browser an.

Die Stärke von PHP liegt jedoch nicht in der Ausgabe von Daten, sondern in der Eingabe von Informationen z.B. in Formularen durch den Benutzer und deren weitere Verarbeitung im Zusammenhang mit Datenbanksystemen, wie z.B. MySQL.

### **A.5.2 PHP - Funktionsweise**

Ein PHP-Aufruf kann durch den direkten Aufruf der Datei als URL, z.B. „www.website.de/datei.php“, oder z.B. auch durch eine Verlinkung in einem <a> Tag mit dem „href“-Attribut erfolgen. Diese Anfrage wird zuerst an den Webserver geschickt, welcher die PHP-Datei an den „PHP-Parser“<sup>19</sup> weitergibt. Im nächsten Schritt werden alle PHP-Abschnitte interpretiert<sup>20</sup>, und mit ihren Ausgaben ersetzt. Die dadurch entstandene Rückgabe wird danach vom Webserver wieder zum Client geschickt, welcher das Ergebnis in seinem Browser sieht.

---

19 Teil von PHP, welcher Quellcode in ein Format umwandelt, welche vom Computer umgesetzt werden kann

20 unmittelbares Ausführen von Code des Parsers ohne Erstellen von ausführbaren Dateien

Der Code eines PHP-Scripts kann an jeglichen Stellen in das bekannte HTML-Grundgerüst eingefügt werden. Als Beispiel wird hier eine Ausgabe (im XML-Format → „<?php ... ?>“) mit dem Befehl „echo“ hinter dem Paragraphen-Tag im Body-Tag eingefügt.

```
<html>

<body>

<p>Text</p>

<br>

<?php
echo („Testausgabe“);

?>

</body>

</html>
```

*Quellcode 30: Ausgabe eines Textes durch PHP innerhalb eines HTML-Dokuments*

Die Ausgabe dieses Code-Ausschnittes führt zu folgender Ausgabe beim Benutzer:

Text

Testausgabe

*Abbildung 34: Ausgabe der "echo"-Funktion kombiniert mit der Ausgabe von HTML*

Abschnitte von PHP-Quelltext können jedoch auch durch andere „Tags“ gekennzeichnet werden. So bewirken z.B. auch „<script language='php'>“ ... </script> und „<? ... >“ das Ausführen von PHP-Code.

### A.5.3 PHP - Aufbau

PHP besteht im Allgemeinen aus 3 autonomen, also voneinander unabhängigen, Teilen.

Die Grundlage bildet der Interpreter, als welcher standardmäßig die freie „Zend Engine“ fungiert. Diese „interpretiert“ PHP-Code innerhalb einer virtuellen Maschine und überprüft diesen, um erhöhten Sicherheitsanforderungen zu entsprechen. Die Eingabe wird dabei innerhalb einer vom übrigen System abgekapselten Instanz des Interpreters zeilenweise durch festgelegte Assoziationen von Befehlen mit ihrem entsprechenden Maschinencode ersetzt und an den Webserver zurückgegeben. Der Interpreter fungiert sozusagen als Wörterbuch von PHP, mit dem sich die zu einem Befehl zugehörigen Aktionen ermitteln lassen.

Ergänzt wird dieser Sprachkern durch sogenannte „Bibliotheken“ oder auch „Module“, welche PHP um zahlreiche weitere Funktionen ergänzt und Schnittstellen zu anderen Anwendungen und Anwendungsbereichen implementiert<sup>21</sup>.

Ein bekanntes Beispiel einer Erweiterung für PHP ist die MySQL-Bibliothek, die Funktionen und Befehle im Zusammenhang mit der gleichnamigen beliebten Datenbanklösung innerhalb von PHP bereitstellt. Durch diese Anbindungen an verbreitete Systeme und Software können die Barrieren zwischen Desktop- und Webanwendung häufig überwunden werden und diese in ihrem Funktionsumfang auf ein gleiches Level gebracht werden. Webanwendungen werden dadurch beliebter und die Realisierung von umfangreichen Webprojekten wird einfacher bzw. möglich.

Als letzten Bestandteil ist die SAPI-Schnittstelle („Server Application Programming Interface“) von PHP zu nennen, deren Funktion die Verbindung von PHP mit dem jeweiligen Webserver ist.

Eine SAPI ist dabei im Allgemeinen ein Mechanismus, der die Verbindung zweier ansonsten inkompatibler Komponenten ermöglicht. In unserem Fall ist die Komponente auf der einen Seite PHP und auf der anderen Seite ein Webserver, z.B. der Apache Webserver. Beide Komponenten sind in ihrer Funktionsweise autonom, weshalb ohne die Anbindung durch eine SAPI das Zugreifen der Komponenten aufeinander nicht möglich wäre. Diese Anbindung wird durch die Verarbeitung der Kommunikation zwischen beiden Teilen ermöglicht, welche durch die SAPI von PHP erfolgt.

Durch diese individuell anpassbare Schnittstelle wurde es möglich PHP auf verschiedensten Webservern auszuführen und die Verbreitung der Sprache enorm zu verstärken.

#### **A.5.4 PHP – Anwendungsgebiete**

Durch seine einheitliche Unterstützung auf verschiedensten Webservern ist PHP nicht an bestimmte Servertypen gebunden und kann dadurch enorm vielfältig eingesetzt werden.

PHP wird hauptsächlich im Bereich der Webentwicklung verwendet, jedoch sind auch lokale Anwendungsszenarien realisierbar. Bekannte Beispiele dafür sind z.B. das WordPress-Framework, Facebook oder auch Wikipedia bzw. das Media Wiki-Framework.

### **A.6 CSS („Cascading Style Sheets“) (Quellen: [09], [20], [21] und [22])**

#### **A.6.1 CSS - Allgemein**

„Cascading Style Sheets“, kurz „CSS“ ist, wie der Name schon sagt, eine Stylesheet-Sprache, die für die Gestaltung und Proportionierung von HTML-Elementen verantwortlich ist. Es wird in der Webanwendung genutzt um Code von seinen Gestaltungsvorgaben zu trennen und um damit Übersichtlichkeit, sowie einfaches Bearbeiten zu gewährleisten.

#### **A.6.2 CSS - Funktionsweise**

In CSS werden Elemente direkt über bestimmte Merkmale angesprochen bzw. übernehmen ausgehend von ihren Elternelementen Attribute. Dieses Prinzip nennt man Vererbung. Die Identifikation von Elementen kann dabei über unterschiedliche Indikatoren erfolgen. So können entweder Elemente einer Klasse, eines Typs, Elemente einer Klasse bzw. eines Typs innerhalb eines Elements oder auch Elemente mit einem bestimmten Namen bzw. bestimmter ID direkt angesprochen werden.

---

<sup>21</sup> verknüpfen von verschiedenen Programmen ineinander

### A.6.3 CSS - Aufbau

Eine CSS-Datei ist eine externe, aus dem originalen Code ausgelagerte, Datei, welche äußerlich durch die Dateiendung „.css“ zu erkennen ist. Innerhalb der CSS-Codestruktur gibt es einzelne Vorlagen, welche nach der folgenden Syntax erstellt werden. :

```
Selector1 {  
    Eigenschaftsname : Wert ;  
    Eigenschaftsname : Wert ;  
}  
Selector2 {  
    Eigenschaftsname : Wert ;  
    Eigenschaftsname : Wert ;  
}
```

*Abbildung 35: CSS Syntax*

Der Selector ist dabei der Identifikator des zu verändernden Elements und kann unterschiedlich aussehen. Er kann weiterhin durch bestimmte Attribute ergänzt werden, wodurch z.B. eine Vorlage nur aufgerufen wird, wenn sich der Mauszeiger über dem angesprochenen Element befindet, also ein „Hover-Event“ („Schwebe-Ereigniss“) auslöst. Die Eigenschaften, die bearbeitet werden, hängen dabei von den Elementen ab, jedoch sind einige Werte für fast alle Elemente verfügbar. Der Wert des Merkmals ist dabei abhängig von der zu bearbeitenden Eigenschaft. Häufig verwendete Werte sind Zahlen, oder bestimmte vordefinierte Befehle, mit denen ein bestimmtes Verhalten, wie z.B. eine Zentrierung von Text einhergeht. Einige Beispiele für häufige CSS-Vorgaben und Selectoren sind im Folgenden aufgeführt.



```

.Klassenname{
    background-color : green;
    text-color : black;
}

#ID{
    text-align: center;
    border: 1 1 solid black;
}

Elemententyp{
    text-decoration: none;
    height: 100%;
}

Selector:hover{
    transform: rotate(20deg);
}

```

*Abbildung 36: Beispiel für verschiedene CSS Syntax*

## A.6.4 CSS - Anwendungsgebiete

Mit CSS können grundlegende Farbgebungen bis hin zu aufwendigen Animationen realisiert werden. Dadurch findet sich dieses in fast allen Webanwendungen und Websites wieder. Mit einer Standardisierung der verwendeten Befehle durch das W3C („World Wide Web Consortium“) ist die Integration in den Browsern weitgehend vereinheitlicht und die Darstellung in verbreiteten Browsern ähnlich bzw. gleich, so dass der Einsatz von CSS zunehmend uneingeschränkt erfolgen kann.

CSS wird dabei genauso auf kleinen Websites wie in komplexen Webanwendungen verwendet, da der Handlungsrahmen enorm groß ist und die richtige Verwendung der Sprache sehr einfach zu erlernen ist.

## A.7 Javascript (JS) (Quellen: [04], [05] und [06])

### A.7.1 JS – Allgemein

„JavaScript ist eine interpretierte Programmiersprache mit objektorientierten Fähigkeiten“ (JavaScript Das umfassende Referenzwerk – O‘Reilly 5. Auflage)

Der offizielle Name der Sprache lautet „ECMAScript“ (ECMA=European Computer Manufacturer’s Association). JavaScript bezeichnet theoretisch nur die Browsereinbindung von Mozilla und Netscape, wird aber in der Praxis als gängige Bezeichnung der Sprache verwendet.

JavaScript ähnelt sich häufig in Aspekten der Syntax mit Programmiersprachen wie C, C++ oder auch Java. Weiterhin inspirierte z.B. PERL das Grundgerüst von Arrays in JavaScript und die Konzepte der Vererbung basieren auf vorhergegangenen Sprachen wie Self.

Ein wichtiger Aspekt der Skriptsprache ist die lose Typisierung der Sprache, welche die Verwendung von Variablen ermöglicht, welche nicht speziell einem Datentyp zugeordnet wurden. Dadurch und durch die relativ einfache Struktur wird JavaScript oft als einfache Skriptsprache abgetan. Dies ist jedoch nur teilweise richtig, da JavaScript zwar auch für simple Funktionen genutzt werden kann, ebenso gut jedoch auch komplexe Mechanismen, wie z.B. Animationen realisieren kann, welche mehr als nur grobe Kenntnis der Sprache erfordern.

## A.7.2 JS – Funktionsweise

JavaScript ist eine unbeschränkt einsetzbare Programmiersprache und besteht aus dem jeweils auf den Anwendungszweck angepassten JavaScript-Interpreter und der Integration von diesem in den jeweiligen Kontext. Am häufigsten wird jedoch die Web-Version der Sprache verwendet, welche auch als einzige im Rahmen der Lernleistung relevant ist. Auf Grund dessen wird im Folgenden auf die Funktionsweise anderer Versionen verzichtet und besonderes Augenmerk auf die Browser-Versionen gelegt.

JavaScript unterstützt von Grund auf elementare Datentypen, wie z.B. Zahlen, Zeichenketten oder auch Wahrheitswerte. Weiterhin können Arrays, Datumswerte und reguläre Ausdrücke als vordefinierte Objekte verwendet werden. Im Kontext der Webentwicklung wird JavaScript um verschiedene Objekte erweitert, welche die Schnittstelle zum Browser bilden und dadurch Interaktionen mit den Anwender ermöglichen. Im Web-Kontext ist der Kern von JavaScript also nur die Grundlage der Sprache und erst durch die Integration in den Browser durch das DOM („Document Object Model<sup>22</sup>“) wird die Manipulation, Darstellung, ö.Ä. von Inhalten ermöglicht. Dabei wird der Code jedoch nur lokal ausgeführt und kann nur über Umwege, z.B. mit Hilfe von AJAX, mit dem Server kommunizieren.

Wird Code beim Anwender ausgeführt, erfolgt dies in einer abgekapselten Umgebung, einer „Sandbox“, um vor dubiosen Code von bösartigen Websites, welcher ansonsten unkontrolliert über Webdokumente an den Anwender übergeben werden könnte, zu schützen. Das Ergebnis der lokalen Laufzeit wird dem Benutzer anschließend ausgegeben und lässt innerhalb dieser Laufzeit Interaktionen zu. Der Funktionsrahmen von JavaScript ist dabei jedoch immer vom verwendeten Browser und dessen Einstellungen abhängig. So können zwar die meisten Browser die standardisierten Befehle ausführen, jedoch besitzen Browser teilweise verschiedene Funktionen, welche nur innerhalb der eigenen Umgebung verfügbar sind bzw. nicht von allen Browser unterstützt werden.

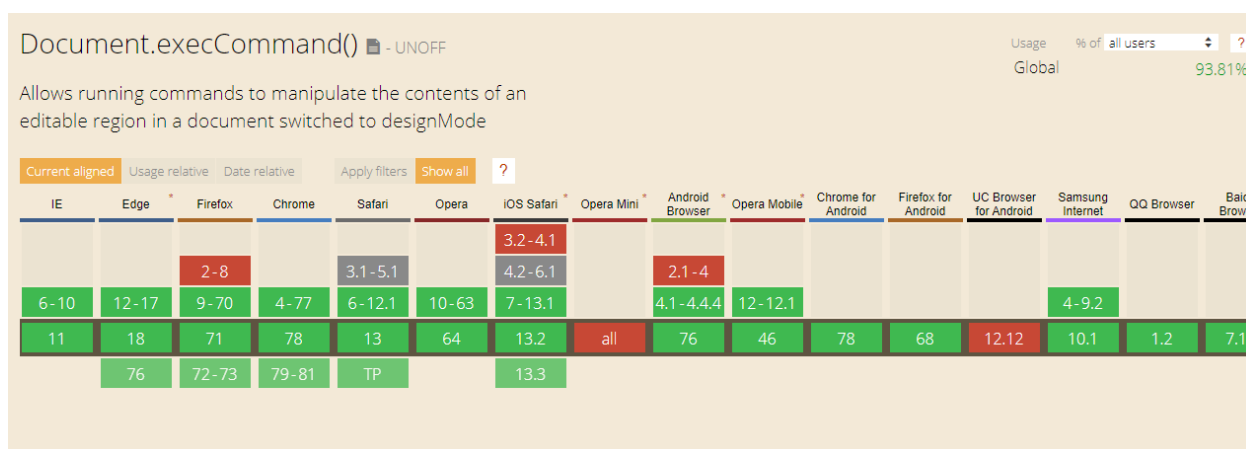


Abbildung 37: schematische Darstellung der Browserunterstützung einer JavaScript-Funktion

Als Beispiel sei hier die `Document.execCommand()`-Funktion von JavaScript genannt, welche laut dieser Quelle z.B. im „UC Browser for Android“ nicht unterstützt wird.

Weiterhin können verschiedene Funktionen z.B. der Zugriff auf das Mikrofon eines Anwenders durch den Browser unterbunden werden.

### A.7.3 JS – Aufbau

JavaScript wird im Web Kontext innerhalb eines HTML-Dokumentes geschrieben bzw. in dieses eingebunden. Beim ersten Fall erfolgt die Abgrenzung von anderem Code durch einen `<script>`-Tag, welcher ähnlich wie die meisten HTML-Tags nach dem Code wieder durch ein `</script>`-Tag geschlossen wird.

```
<html>
<head>
<title>Beispiel</title>
</head>
<body>
<p id="Testelement">Hallo</p>
<script>
var element = document.getElementById;
element.backgroundColor="green";
</script>
</body>
</html>
```

*Quellcode 31: in HTML Kontext integrierter JavaScript-Code*

Bei diesem Beispiel erfolgt die Einbindung von JavaScript direkt im Dokument. Das Ergebnis der Anweisung würde in diesem Fall ein Grünfärbung der Hallo-Ausgabe sein.

Ähnlich wird im zweiten Fall JavaScript verwendet. Der einzige Unterschied besteht in der Auslagerung von Code in ein anderes Dokument um z.B. Übersichtlichkeit oder Struktur eines Projektes zu gewährleisten. Die Einbindung von ausgelagertem Code erfolgt dabei über das „src“-Attribut des Script-Tags, welches die Quelle des Skriptes bestimmt und damit die Verknüpfung beider Dokumente ermöglicht.

```

<html>
  <head>
    <title>Beispiel</title>
    <script src="script.js"></script>
  </head>
  <body>
    <p id="Testelement">Hallo</p>
  </body>
</html>

```

*Quellcode 32: beispielhafter Aufbau eines HTML-Dokuments mit externem JavaScript-Code*

```

var element = document.getElementById;
element.backgroundColor="green";

```

*Quellcode 33: externer JavaScript-Code, welcher im HTML-Dokument ausgeführt wird*

#### A.7.4 JS - Anwendungsgebiete

JavaScript wird am häufigsten im Kontext der Webprogrammierung und in den entsprechenden Browsern verwendet. Weiterhin sind der Sprache jedoch keine Grenzen gesetzt und praktisch jedes Programm kann mit Hilfe eines JavaScript-Interpreters die Sprache innerhalb anderer Kontexte nutzen. Einige Beispiele dafür sind z.B. die Verwendung von JavaScript im Scripting Host von Windows, innerhalb der Scripting-Funktionalität des Adobe Flashplayers<sup>23</sup> oder auch als Zusatzmodul für jegliche Java Anwendung über den Anbieter Oracle<sup>24</sup>.

Viele JavaScript Interpreter sind außerdem frei verfügbar und können dadurch problemlos in eigene Projekte eingebunden werden.

#### A.7.5 JS – jQuery

Durch die steigende Nachfrage nach immer aufwendigeren Webanwendungen und Funktionen innerhalb immer kürzer werdenden Zeitrahmen nimmt auch die Nachfrage nach Bibliotheken für JavaScript zu. Diese Bibliotheken erleichtern verschiedene Abläufe innerhalb der Sprache durch vordefinierte Routinen und die optimale Nutzung von Parametern. Eine der weltweit bekanntesten Bibliotheken für JavaScript ist dabei jQuery, welche durch ihren enormen Umfang an Funktionen und Möglichkeiten fast schon als separate Programmiersprache behandelt werden könnte.

<sup>23</sup> Software zur Anzeige von Multimediainhalten

<sup>24</sup> Soft- und Hardwarehersteller aus den USA

Ein einfaches Beispiel für die Optimierung von Abläufen und Quellcode durch jQuery wird im folgenden Vergleich zwischen normalem JavaScript und JavaScript mit der Erweiterung durch jQuery ersichtlich. Dabei haben beide Codeabschnitte das gleiche Ergebnis, unterscheiden sich jedoch in Länge und Übersichtlichkeit des Quellcodes. Das Ergebnis des folgenden Beispiels ist weiterhin das Hinzufügen von zwei Buttons zum „Body-Element“ und das Hinzufügen eines „Event-Listener“<sup>25</sup> zu jeweils einem Button.

---

25 Bestandteil von Programmen, welche auf ein bestimmtes Ereignis, wie z.B. eine bestimmte Uhrzeit, warten und erst beim Erfüllen der Bedingung ausgeführt werden

```

<html>
  <head>
    <script src="http://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-
CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFIBw8HfCJo="
crossorigin="anonymous"></script>
  </head>
  <body id="inhalt"></body>
  <script>
    //////////////////////////////////////
    Einfügen von HTML über JavaScript
    //////////////////////////////////////

    //Lösung mit "normalem" JavaScript
    document.getElementById("inhalt").innerHTML += "<button class='knoepfe' id='knopf-
1'>Knopf 1</button>";

    //Lösung mit jQuery
    $("#inhalt").append("<button class='knoepfe' id='knopf-2'>Knopf 2</button>");

    //////////////////////////////////////
    Hinzufügen von EventListener
    //////////////////////////////////////

    //Lösung mit "normalem" JavaScript
    document.getElementById("knopf-1").addEventListener("click", function() {
      alert("Auf Knopf 1 geklickt.");
    }, false);
    //Lösung mit jQuery
    $("#knopf-2").click(function() {
      alert("Auf Knopf 2 geklickt (Methode 1).");
    });

  </script>
</html>

```

*Quellcode 34: beispielhafter Vergleich von JavaScript mit jQuery*

## A.8 AJAX („Asynchronous JavaScript and XML“) (Quellen: [06] und [16])

### A.8.1 AJAX – Allgemein

AJAX ist eine Datenübertragungstechnik, welche innerhalb von Webanwendungen zur Informationsbereitstellung während der Laufzeit genutzt wird. Dadurch werden Ladezeiten verkürzt und die Performance verbessert sich. In den letzten Jahren wurde die Technik weitestgehend standardisiert und konnte dadurch eine hohe Kompatibilität und Verbreitung erreichen.

### A.8.2 AJAX – Funktionsweise

Im Rahmen der Lernleistung werden Ajax-Anfragen mit der JavaScript-Bibliothek jQuery (vgl. Kap.: A.7.5 JS – jQuery S. 60) generiert. Die angeforderten Daten werden dadurch direkt in ein JSON-Format<sup>26</sup> konvertiert, welches als räumliches Objekt, als Array, verwendet werden kann. Ein Beispiel für die Darstellung von Informationen durch JSON ist im Nachfolgenden gegeben. Dort wird ein Array mit 4 Feldern dargestellt und die Inhalte der einzelnen Felder strukturiert angezeigt.

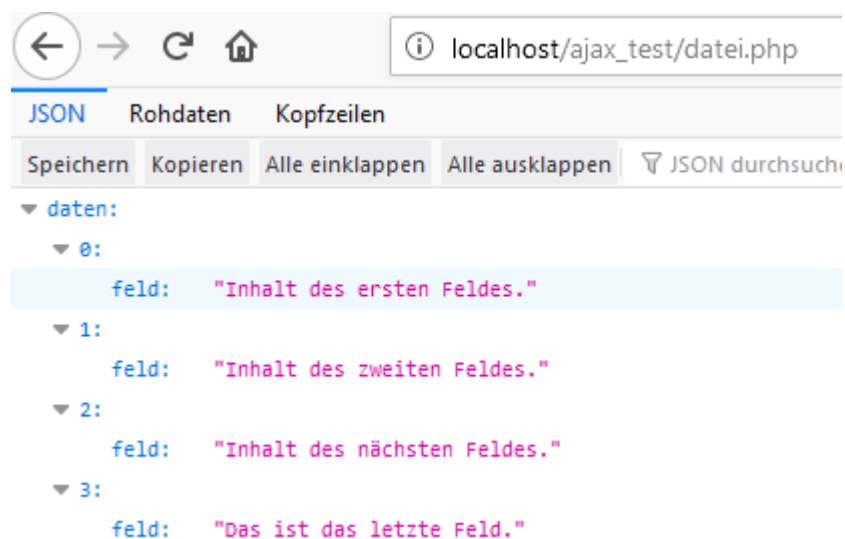


Abbildung 38: Darstellung von Informationen im JSON-Format

Im zweiten Beispiel wird die Funktionsweise von Ajax demonstriert. Zuerst wird in der HTML-Datei eine Ajax-Anfrage an die PHP-Datei geschickt. Diese Anfrage liefert das in der PHP-Datei definierte Objekt zurück. Bei Erfolg der Ajax-Anfrage wird anschließend das Auslesen und die Darstellung des Objekts getriggert. Bei einem Fehler in der Anfrage wird dieser ausgegeben.

```

<?php
header("Content-Type: application/json");

$json_return = array();
$json_return["daten"][0]["feld"] = "Inhalt des ersten Feldes.";
$json_return["daten"][1]["feld"] = "Inhalt des zweiten Feldes.";
$json_return["daten"][2]["feld"] = "Inhalt des nächsten Feldes.";
$json_return["daten"][3]["feld"] = "Das ist das letzte Feld.";

exit(json_encode($json_return));
?>

```

Quellcode 35: PHP-Datei "datei.php" erstellt Array und gibt dieses zurück

```

<html>
  <head>
    <title>AJAX Test</title>
    <meta charset="utf-8"/>
    <script src="http://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-
CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
crossorigin="anonymous"></script>
  </head>
  <body id="inhalt"></body>
  <script>
    $.ajax({
      url: "datei.php",
      cache: false,
      success: function(data) {
        $("#inhalt").html("");
        for(var i = 0; i < data.daten.length; i++) {
          $("#inhalt").append("<p>" + data.daten[i].feld + "</p>");
        }
      },
      error: function(jqXHR, textStatus, errorThrown) {
        alert("Beim Ausführen der AJAX-Anfrage trat ein Fehler auf: " + errorThrown);
      }
    });
  </script>
</html>

```

Quellcode 36: HTML-Datei "index.html" führt AJAX-Anfrage aus und stellt das übergebene Array dar



### A.8.3 AJAX – Aufbau

Ajax besitzt keinen direkten Compiler, wie z.B. HTML, sondern schickt eine HTTP-Anfrage an einen Server. Die Antwort wird dann vom Server zurückgegeben und kann durch Skripte direkt verwendet werden. Diese Informationsbeschaffung läuft dabei „asynchron“, also parallel zur Webanwendung, ab. Dadurch kann die Anwendung weiterhin verwendet werden und es muss nicht auf das Laden der Daten vom Server gewartet werden.

### A.8.4 AJAX – Anwendungsgebiete

AJAX wird heutzutage in vielen cloudbasierten Programmen verwendet. Die Anwendungsgebiete reichen dabei von Handyspielen, wie z.B. Clash of Clans, bis hin zu Websites, wie z.B. Instagram.

## A.9 SQL (Quelle: [02])

### A.9.1 SQL - Allgemein

SQL ist laut „[db-engines.com/en/ranking](https://db-engines.com/en/ranking)“ (zuletzt abgerufen am 11.02.2020) eine der am meisten verwendeten Datenbanksprachen der Welt und damit die häufigste Schnittstelle zwischen Datenbankserver und Anwender. Durch ihre einfache Syntax und die Verwendung von relationalen Prinzipien<sup>27</sup> ist SQL sehr beliebt und kommt in vielen Datenbanksystemen vor.

### A.9.2 SQL – Funktionsweise

Bei SQL wird eine Anweisung direkt in der jeweils verknüpften Datenbank bzw. auf dem Datenbankserver ausgeführt. Dabei muss z.B. beim Auslesen auf die eindeutige Identifizierung eines Datensatzes geachtet werden, da sonst mehrere Ergebnisse ausgegeben werden, welche allesamt die vorgegebenen Eigenschaften erfüllen. Eine weitere Besonderheit ist das Format der Ausgabe. Diese ist keine Variable, sondern ein räumliches Objekt, eine „Ergebnismenge“. Man kann sie sich als eine Art Tabelle vorstellen, welche bereits in ihrer, durch eine Anweisung erzeugten, Form z.B. als neue SQL-Tabelle abgespeichert werden könnte oder auch in weiter verschachtelten Abfragen als Eingabe genutzt werden kann.

```
SELECT * FROM `metadata`
```

id	name	metadata
1	title	GutsMuths Gymnasium Quedlinburg
2	description	Website des GutsMuths Gymnasium Quedlinburg
3	revisit	30
4	index	INDEX

4 Datensätze (0.000 s) Bearbeiten, Explain, Exportieren

Abbildung 39: Ausgabe der SQL-Anweisung "SELECT \* FROM `metadata`" in Tabellenform

### A.9.3 SQL – Aufbau

SQL setzt sich im Allgemeinen aus 2 unterschiedlichen Arten von Anweisungen zusammen. Dazu zählen Schemaanweisungen und Datenanweisungen. Ein Beispiel für weitere Anweisungen wäre z.B. eine Transaktionsanweisung, welche im Kontext der Schulwebsite jedoch nicht weiter behandelt wird.

<sup>27</sup> Speicherung von Daten in separaten Tabellen und anschließende Verknüpfung über „Relationen“

Grundlage von SQL sind dabei Schemaanweisungen, welche für die Struktur der Daten verantwortlich sind. Mit ihnen lassen sich z.B. Tabellen und verschiedene Spalten mit vordefinierten Datentypen erstellen.

```
CREATE TABLE beitraege (  
    id INT NOT NULL PRIMARY KEY,  
    name VARCHAR(40) NOT NULL  
);
```

*Quellcode 37: beispielhafte Schemaanweisung zum Erstellen einer neuen Tabelle mit zwei verschiedenen Datentypen*

Um in diese Struktur Daten einzubinden werden Datenanweisungen verwendet. Mit ihnen lassen sich z.B. Daten in vorher erstellte Tabellen/Strukturen einfügen oder auch Daten mit bestimmten Parametern ausgeben.

```
INSERT INTO beitraege (id, name) VALUES (1, 'Startseite');
```

*Quellcode 38: beispielhafte Schemaanweisung zum Einfügen von Daten in eine Tabelle*

#### Legende

Türkis	SQL-Anweisung
Rot	selbst vergebene Namen von Tabelle und Spalten
Grün	Datentyp + (falls benötigt) Maximalwert
Violett	Eigenschaften: <ul style="list-style-type: none"><li>• NOT NULL = Datenfeld darf nicht leer sein</li><li>• PRIMARY KEY = legt das Datenfeld als Primärschlüssel fest</li></ul>
orange	Daten

### A.9.4 SQL – Anwendungsgebiete

Im Allgemeinen kann man sagen, dass SQL fast in allen Bereichen der Programmierung vertreten ist, die Daten speichern und Strukturieren müssen. Basierend auf SQL wurden deswegen für unterschiedlichste Anwendungsszenarien verschiedenste Systeme geschaffen. Einige weitverbreitete Beispiele sind dabei SQLite, MySQL oder auch MSSQL.

SQL findet meist in der Webprogrammierung Anwendung. So basieren Datenbanksysteme wie z.B. MySQL auf SQL bzw. werden durch diese ergänzt. Weitere Anwendungsgebiete liegen unter anderem in Datenverwaltungssystemen von Datenzentren, Konzernen oder auch Programmen des .Net-Frameworks, welches basierend auf SQL ein eigenes Datenbanksystem MSSQL zur Verwaltung von Daten nutzt. Eine freie Version eines Datenbanksystems für Programme ist z.B. SQLite, welches als Bibliothek frei verfügbar ist und basierend auf SQL ein relationales Datenbanksystem einschließt.

## **A.10 MYSQL (Quellen: [03] und [08])**

### **A.10.1 MYSQL – Allgemein**

MYSQL ist die am weitesten verbreitete frei verfügbare Datenbank der Welt. Sie arbeitet als Server auf Grundlage der Datenbanksprache SQL (vgl. Kap. A.9 SQL (Quelle: [02]) S. 65) und bietet viele Vorteile. Durch einfache SQL-Befehle und die umfassende Dokumentation der Sprache im Internet und in Büchern kann der Umgang mit MySQL leicht erlernt werden. Weiterhin ist MySQL auf allen gängigen Plattformen verfügbar und kann nicht nur im Web, sondern auch in Anwendungen vieler Programmiersprachen verwendet werden. Der größte Vorteil ist jedoch die enorme Schnelligkeit der Datenbank gekoppelt mit einer geringen Ausfallrate und hoher Stabilität

### **A.10.2 MYSQL – Funktionsweise**

MySQL ist ein typisches Client/Server-System, welches aus einem Datenbank-Server und mindestens einem Anwender/Client besteht. Diese Clients können im Allgemeinen z.B. durch SQL-Befehle (siehe Kap. A.9 SQL (Quelle: [02]) S.65) auf die im Datenbank-Server gespeicherten Daten zugreifen bzw. Daten auf diesem speichern.

Dabei wird diese Interaktion zwischen Client und Server meist mit Hilfe einer Datenbank-Oberfläche innerhalb eines Browsers oder als Desktop-Anwendung bzw. direkt innerhalb eines Systems, wie z.B. einer dynamischen Website, durch automatisierte Vorgänge realisiert. Die Schnittstelle zwischen Server und Client ist dadurch am häufigsten durch eine PHP Bibliothek gegeben (vgl. Kap. A.5.3 PHP - Aufbau S.54).

### **A.10.3 MYSQL - Anwendungsgebiete**

MySQL wird heutzutage durch die freie Verfügbarkeit in fast allen Webanwendungen genutzt. Des weiteren werden jedoch z.B. auch viele Firmennetzwerke mit einem MySQL-Server gekoppelt. Durch die Anbindung eines einzigen Datenbanksystem können Daten dabei einfach und schnell auch über größere Infrastrukturen hinweg genutzt werden.

So könnten z.B. in Schulen die Schülerdaten an einer zentralen Stelle aufbewahrt, jedoch auf jeglichen an das System angeschlossenen Clients abgerufen werden. Weiterhin könnte eine Schnittstelle über einen Server den Zugriff von Clients außerhalb des Netzwerkes auf die Daten ermöglichen. Eine solche Einbindung eines MySQL-Systems könnte dadurch zu einer Erleichterung von Arbeitsabläufen führen.