

**PROGRAMMIEREN EINER INTERAKTIVEN SCHULBUCH-BÖRSE FÜR
ZÜRCHER KANTONSSCHÜLER**

Eine Maturitätsarbeit an der
KANTONSSCHULE LIMMATTAL

vorgelegt von
MARC PHILIPP MEINERS

Klasse M6a
im Fach Informatik

betreut von
Dr. Harald Pierhöfer

Abstract

Die vorliegende Maturitätsarbeit zeigt den Prozess der Programmierung einer Schulbuch-Börse auf. Dabei wurde in Form einer Website ein Tool geschaffen, das es Zürcher Kantonsschülern erlauben soll, nicht mehr benötigte Schulbücher zu verkaufen und gebrauchte Exemplare zu erwerben. Die Internetseite ist vollkommen interaktiv und automatisiert: Sie verfügt über eine Registrierungs- und Login-Lösung sowie über ein benutzerfreundliches Dashboard, in dem Bücher und Benutzerdaten verwaltet werden können. In der Bücher-Übersicht können die zum Verkauf stehenden Bücher eingesehen, gefiltert sowie schlussendlich gekauft werden. Der Buch-Registrierungsprozess wird ergänzt durch eine Schnittstelle zu einer externen Plattform, sodass viele Buch-Informationen nur mithilfe der ISBN-Nummer bereits automatisch abgerufen werden können und nicht vom Verkäufer manuell eingegeben werden müssen. Die Überprüfung der Schul-E-Mail-Adresse sowie ein ausgeklügeltes Reaktivierungssystem stellen ausserdem sicher, dass bei der Plattform nur Personen registriert sind die aktuell oder vor nicht allzu langer Zeit eine Zürcher Kantonsschule besuchen oder besucht haben. Für den Schutz vor allfälligen Attacken auf die Website wurden Vorkehrungen getroffen, welche Eingaben des Benutzers nicht nur auf Form und Länge, sondern auch auf das Beinhalten von Programmcode überprüfen, der beispielsweise über Formulare eingeschleust werden könnte.

Inhaltsverzeichnis

Abstract.....	i
Inhaltsverzeichnis.....	ii
1 Einleitung	1
1.1 Idee	1
1.2 Ziel.....	1
1.3 Motivation	1
1.4 Werkzeuge	1
1.4.1 HTML	1
1.4.2 CSS	1
1.4.3 PHP.....	2
1.4.4 SQL / MySQL	2
1.4.5 XAMPP	2
1.4.6 Adobe Dreamweaver 2019.....	3
1.5 Vorkenntnisse	3
1.6 Anmerkungen für den Leser	3
2 Die Website.....	4
2.1 Grundaufbau der Website	4
2.1.1 Ordner – und Filestruktur	4
2.1.1.1 Index.php	5
2.1.1.2 Header.php.....	5
2.1.1.3 Footer.php	6
2.1.1.4 Functions.php	6
2.1.1.5 Sessions	6
2.1.1.6 Style.css	7
2.1.2 Datenbankstruktur.....	7
2.1.2.1 Die Tabelle «benutzer».....	8
2.1.2.2 Die Tabelle «buecher»	8
2.2 Registrierungsverfahren.....	9
2.2.1 Registrierungsformular	9
2.2.1.1 Drop-Down-Menüs des Formulars	9
2.2.2 Bedingungen.....	10
2.2.3 Überprüfung der E-Mail-Adresse.....	10
2.2.4 Behandlung des Passworts	10
2.2.5 Datenbankeintrag	11
2.2.6 E-Mail-Bestätigung und Aktivierung des Kontos.....	11
2.3 Login-Verfahren.....	12

2.3.1	Formular-Auswertung	12
2.4	Reaktivierungssystem	13
2.4.1	Reaktivierungsprozess	13
2.5	Bücherbörse	14
2.5.1	Buch-Registrierung	14
2.5.1.1	ISBN-Nummer	14
2.5.1.2	Google Books API	15
2.5.1.3	Buch-Registrierungsformular	16
2.5.1.4	Auswertung des Buch-Registrierungsformulars	17
2.5.2	Bücher-Übersicht	18
2.5.2.1	Filtermöglichkeiten	18
2.5.2.1.1	Filter-Sidebar	18
2.5.2.1.2	Hintergrund-Technik des Filters	20
2.5.2.2	Buchrahmen	21
2.5.2.3	Blättermöglichkeit	22
2.5.3	Buchdetails-Ansicht	23
2.5.4	Kaufprozess	25
2.6	Dashboard	26
2.6.1	Benutzerdaten	26
2.6.1.1	Benutzerdaten-Änderung	27
2.6.2	Bücherverwaltung	28
2.6.2.1	Sidebar	28
2.6.2.2	Grundprinzip der fünf Kategorien	28
2.6.2.3	Hintergrund-Technik	30
2.6.2.3.1	Datenbankabfragen	30
2.6.2.3.2	Aktionen in der Bücherverwaltung	31
2.7	Schutz vor SQL-Injections	31
3	Schlusswort	32
3.1	Zusammenfassung	32
3.2	Schwierigkeiten	32
3.3	Mögliche Weiterentwicklungen des Projekts	32
3.4	Erkenntnisse aus der Arbeit	33
4	Literaturverzeichnis	34
5	Abbildungsverzeichnis	35
6	Einhaltung rechtlicher Vorgaben	35

1 Einleitung

1.1 Idee

Ursprünglich wollte ich als Maturitätsarbeit ein elektronisches Klassenbuch-System programmieren. Leider musste ich erfahren, dass ein solches System für unsere Schule bereits in Arbeit war. Mein Betreuer hatte allerdings eine andere Idee. Sein Vorschlag war eine schulinterne Schulbuch-Börse für die Kantonsschule Limmattal. Nach einigen Überlegungen erkannte ich, dass er damit eine Art Marktlücke entdeckte. Ich habe seine Idee als Grundlage für meine Maturitätsarbeit genommen und diese ausgeweitet beziehungsweise ergänzt.

1.2 Ziel

Mein Ziel ist es, eine Website zu programmieren, bei der sich Schülerinnen und Schüler aller Zürcher Kantonsschulen registrieren können, um ihre Bücher zu verkaufen und gebrauchte Exemplare zu erwerben. Die Internetseite soll ausserdem vollkommen interaktiv sein und es sollen so gut wie keine Eingriffe seitens des Administrators nötig sein.

1.3 Motivation

Gerade die aktuellen Klimadiskussionen zeigen auf, wie unsere Konsum- oder auch «Wegwerf»-Gesellschaft funktioniert. Bezogen auf das Thema meiner Arbeit bedeutet dies einen sich alljährlich für die Schülerinnen und Schüler wiederholenden Kaufzwang von neuen Schulbüchern. Die Exemplare werden meist nur einige Monate gebraucht und liegen dann, oftmals im Keller oder Estrich, in irgendeiner Kiste, bis sie schlussendlich irgendwann im Altpapier landen. Mit meiner Maturitätsarbeit will ich diesen Verfahrensweisen entgegentreten und ein Medium schaffen, das allen Beteiligten, darunter auch der Umwelt, eine Win-win-Situation beschert.

1.4 Werkzeuge

HTML, CSS, PHP und SQL sind die Programmier-, Auszeichnungs- und Datenbanksprachen, mit denen ich mein Web-Projekt realisiere. Diese vier Werkzeuge stehen in engem Zusammenhang, wobei jedes seinen eigenen Zuständigkeitsbereich abdeckt. Die Sprachen gehen ineinander über und bilden dank zahlreicher Schnittstellen ein ganzes Konstrukt, mit dem fast jede Art von Website realisierbar ist. SQL wird in Form von MySQL, einem Datenbankverwaltungssystem, verwendet. Die Website wird auf einem Apache 2 Server unter Linux laufen.

1.4.1 HTML

HTML (Hypertext Markup Language) bildet als maschinenlesbare Sprache die Grundlage zur Strukturierung von modernen Websites [38]. HTML als Programmiersprache zu bezeichnen wäre falsch, weil damit keine Algorithmen und Befehlsstrukturen realisierbar sind. Vielmehr ist das Schreiben in HTML mit der Erstellung von Worddokumenten vergleichbar.

1.4.2 CSS

So wie HTML für die Strukturierung zuständig ist bildet CSS (Cascading Style Sheets) die Basis zur Gestaltung von Websites [39]. Es können damit beispielsweise Farben, Formen und Abstände von HTML Elementen bestimmt werden.

1.4.3 PHP

PHP ist eine Programmiersprache, die als freie Software verbreitet wird und an C und Pearl angelehnt ist. Sie dient dem Bau von dynamischen Weblösungen und ist sehr verbreitet in der Webentwicklung [40]. PHP ist eine serverbasierte Sprache, was heissen soll, dass die Ausführung der Programme auf dem Server geschieht und dem Endnutzer beziehungsweise dessen Browser nur der entsprechend auf die Anfrage des Clients (=Endnutzer) angepasste HTML Code übermittelt wird (siehe Abb. 1). Das hat mehrere Vorteile: Einerseits werden beispielsweise keine Berechnungen auf dem Endgerät des Benutzers ausgeführt, die dessen Gerät auslasten könnten, andererseits kann der genaue PHP Code geheim gehalten werden, sodass der Code nicht unerlaubt kopiert werden kann.

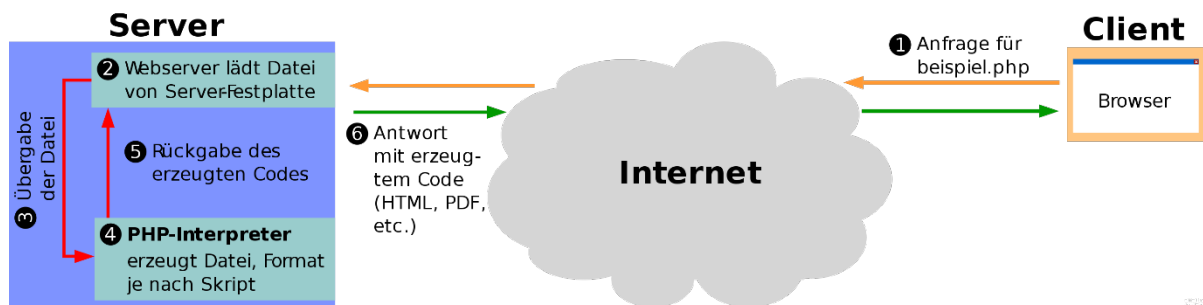


Abb. 1: Datenübermittlung zwischen Client und Server unter Verwendung von PHP [49]

1.4.4 SQL / MySQL

MySQL dient meist der Datenspeicherung für Webservices und wird oft im Zusammenspiel mit PHP verwendet. Es handelt sich um ein Datenbankverwaltungssystem, welches ebenfalls kostenlos verfügbar ist [41]. SQL (Structured Query Language) ist die Datenbanksprache, mit der MySQL Datenbanken bearbeitet und Datenstrukturen definiert werden können [42]. In Datenbanken können Daten ähnlich wie bei einer Tabelle strukturiert abgespeichert werden.

1.4.5 XAMPP

XAMPP ist ein Programmpaket des Projekts Apache Friends. Es ist für Web-Entwickler gedacht, die sich möglichst einfach ein lokales Serversystem zu Testzwecken installieren möchten [43]. Für meine Arbeit benötige ich folgende Software, die bei XAMPP integriert ist und miteinander verwendbar ist:

- Apache
- MySQL
- Mercury
- PHP
- phpMyAdmin

Apache, MySQL und Mercury sind Serversysteme. Apache wird als Webserver verwendet, MySQL als Datenbank-Server und Mercury als Mail-Server [7]. Die lokale Testumgebung hat, wie am Namen erkennbar, den Vorteil, dass sie eben lokal ist und somit keine Dritte auf die Website zugreifen können, sofern der PC nicht von Viren befallen ist.

Während PHP den Code interpretiert, dient phpMyAdmin der Datenbankverwaltung. Es können damit beispielsweise Datenbankstrukturen bestimmt und Datenbankeinträge ausgelesen werden. Es handelt sich um ein webbasiertes Tool, das über den lokalen Webserver als grafische Umgebung im Browser aufgerufen werden kann.

Anzeigen	Struktur	SQL	Suche	Einfügen	Exportieren	Importieren	Rechte	Operationen	Nachverfolgung	Trigger
Tabellenstruktur		Beziehungsansicht								
#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion	
<input type="checkbox"/>	1 id	int(11)			Nein	kein(e)		AUTO_INCREMENT	Bearbeiten	Löschen Mehr
<input type="checkbox"/>	2 datum_erstellt	datetime			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	3 datum_aktualisiert	datetime			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	4 datum_verkauft	datetime			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	5 datum_verkauf_abgeschlossen	datetime			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	6 benutzer_id	int(11)			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	7 benutzer_geloescht	int(11)			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	8 gekauft_von_id	int(11)			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	9 schule	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	10 isbn	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	11 cover_link	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	12 buch_titel	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	13 autor	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	14 jahr	int(11)			Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	15 verlag	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	16 kategorie	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	17 zustand	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr
<input type="checkbox"/>	18 preis	tinytext	utf8_general_ci		Nein	kein(e)			Bearbeiten	Löschen Mehr

☐ Alle auswählen
 markierte: ☐ Anzeigen ☐ Bearbeiten ☐ Löschen ☐ Primärschlüssel ☐ Unique ☐ Index ☐ Volltext ☐ Zu zentralen Spalten hinzufügen
☐ Zentrale Spalten entfernen

Abb. 2: Struktur-Ansicht der "buecher" Tabelle in phpMyAdmin (Eigenquelle)

1.4.6 Adobe Dreamweaver 2019

Adobe Dreamweaver ist ein Code-Editor des Unternehmens Adobe Inc. zur Realisierung moderner Websites. Das Programm bietet neben der normalen Funktion als Editor auch die Möglichkeit der grafischen Bearbeitung von Websites sowie Autovervollständigungen des Codes [45].

1.5 Vorkenntnisse

In der Vergangenheit habe ich bereits eine Website mit den zuvor erwähnten Sprachen und Tools erstellt, weshalb ich ein gewisses Grundwissen mitbringe. Des Weiteren verfüge ich über allgemeine Programmiererfahrung, da wir in den Schwerpunktfächern Mathematik und Physik einige praktische Projekte mit der Programmiersprache «Matlab» realisiert haben.

1.6 Anmerkungen für den Leser

Der Hauptteil meiner Maturitätsarbeit besteht aus dem Schreiben des Programm-Codes für die Website. Deshalb ist es mir wichtig, an dieser Stelle zu erwähnen, dass in der hier vorliegenden schriftlichen Arbeit nur Ausschnitte aus ebendiesem Code ersichtlich sind. Zum besseren Verständnis der einzelnen Themen und zur Kenntnisnahme meiner eigentlichen Arbeit ist es daher zentral, den digital abgegebenen Programm-Code zu Rate zu ziehen – bestenfalls während des Lesens der schriftlichen Arbeit.

Die im folgenden Kapitel ersichtlichen Ausschnitte des Codes sind *an ihrer blauen Farbe und speziellen Formatierung* erkennbar. Sie enthalten oftmals in Form von Kommentaren wichtige Erläuterungen zur aufgezeigten Thematik und sind wie der normal formatierte Fliesstext zu betrachten und mit einzubeziehen.

Der Name beziehungsweise die Domain der Website lautet «kantinet.ch». Die Seite ist zum Zeitpunkt der Abgabe der schriftlichen Arbeit allerdings nicht online abrufbar.

Die Datei «nutzungsbedingungen.php» ist bereits in die Website implementiert, enthält allerdings noch keinen Inhalt und wird in dieser Arbeit nicht behandelt. Vor einer allfälligen

Veröffentlichung der Website wird die Datei mit aktuellen rechtlichen Bedingungen versehen.

2 Die Website

2.1 Grundaufbau der Website

Die zur Funktion und Darstellung der Website zuständigen Dateien befinden sich alle in einem zentralen Ordner. Dabei handelt es sich um mehrere PHP-Dateien (Endung .php), eine CSS-Datei (Endung .css) und eine Bilddatei (Endung .png). Der HTML-Code und allfällige SQL-Befehle sind direkt in die PHP-Dateien integriert. Die Datenbank, welche nicht im erwähnten zentralen Ordner gespeichert ist, ist in zwei Tabellen eingeteilt: es gibt die «benutzer»- und die «buecher»-Tabelle. Der Speicherort befindet sich serverintern in einem Ordner des Programmes MySQL.

2.1.1 Ordner – und Filestruktur

Der Ordner der Website setzt sich folgendermassen zusammen:

- aktivierung.php
- buch.php
- buecherboerse.php
- dashboard.php
- footer.php
- functions.php
- header.php
- index.php
- login.php
- nutzungsbedingungen.php
- reaktivierung.php
- registrieren.php
- standard-cover.png
- style.css

Die Dateien «header.php», «footer.php», «functions.php» und «style.css» werden vom Client nicht direkt aufgerufen. Vielmehr sind es Dateien, die Eigenschaften und Funktionen enthalten, die auf mehreren Unterseiten der Website verwendet werden. Auf diesen Seiten werden die genannten Dateien eingebunden. Das hat den Vorteil, dass beispielsweise bei einer Änderung des Kopfbereiches der Website, die auf fast allen Unterseiten angezeigt wird, nicht jede dieser Seiten einzeln bearbeitet werden muss. Die Änderung müsste in diesem Falle nur in der Datei «header.php» geschehen. Die genannten Files werden in den nun folgenden Kapiteln genauer beschrieben.

Die Eigenschaften und Funktionsweisen der übrigen «Hauptdateien» werden später aufgezeigt. Deren Grundstruktur setzt sich folgendermassen zusammen:

Code-Ausschnitt 1: Vorkommen in mehreren Dateien

```
<?php include'functions.php'; ?>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="style.css" />
    <title>TITEL DER UNTERSEITE</title>
  </head>
  <body>
    <div class="wrapper">
      <?php include'header.php'; ?>
```



```

        <div class="maincontent">
            <div class="inhalt">
                INHALT DER UNTERSEITE
            </div>
        </div>
    <?php include 'footer.php'; ?>
</div>
</body>
</html>

```

Dieser Code-Ausschnitt zeigt, wie die Einbindung der Dateien funktioniert und wie die Haupt-Div-Container zusammengesetzt sind. In den folgenden Kapiteln wird genauer darauf eingegangen.

Ausserdem wird die Kodierung auf UTF-8 gestellt. So können Zeichen-Fehler in der Datenbank vermieden werden.

2.1.1.1 Index.php

Die Startseite (index.php) wird angezeigt, wenn die Stamm-Domain (kantinet.ch) aufgerufen wird. Die Seite hat keine besondere Funktion, ist aber sozusagen die erste Anlaufstelle für Clients. Die Seite kann unabhängig davon, ob der Benutzer eingeloggt ist oder nicht, eingesehen werden.

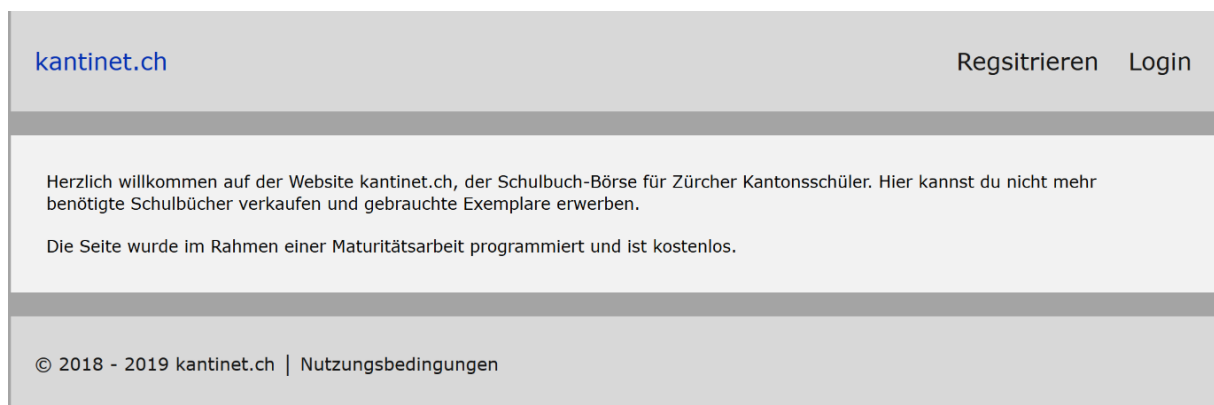


Abb. 3: Startseite (Eigenquelle: index.php)

2.1.1.2 Header.php

Im Folgenden ist der Code der Datei header.php ersichtlich. Die Menüeinträge, die später auf allen Unterseiten der Website ersichtlich sein werden, werden über den <a> Tag als Link angezeigt. Die Darstellung dieser Links und des Headers (Kopfbereich der Website) als Ganzes kann über die «style.css»-Datei gesteuert werden.

Die «Sessions» werden in einem späteren Kapitel erklärt. Sie sorgen dafür, dass dem nicht angemeldeten Benutzer die Links «Bücherbörse», «Dashboard» und «Logout» nicht angezeigt werden und dem eingeloggten Benutzer entsprechend die Menüs «Login» und «Registrieren» nicht ersichtlich sind.

Die sogenannten <div>-Container helfen dabei, der Website eine Struktur zu verleihen. Sie können über «class»-Attribute in der «style.css»-Datei angesprochen werden. So können den Elementen innerhalb eines Containers bestimmte Darstellungs-Einstellungen zugeordnet werden.

Code-Ausschnitt 2: header.php

```

<div class="header">
<div class="headermenu">
    <div class="headermenu-links">
        <a style="color: #0431B4;" href="index.php">kantinet.ch</a>
    
```

```

        <?php
        if (isset($_SESSION['benutzer_id'])){
            ?>
            <a href="buecherboerse.php">Bücherbörse</a>
        <?php } ?>
    </div>
    <div class="headermenu-rechts">
        <?php
        if (isset($_SESSION['benutzer_id'])){?>
            <a href="dashboard.php">Dashboard</a>
            <a href="index.php?logout=1">Logout</a>
        <?php } ?>
        <?php
        if (!isset($_SESSION['benutzer_id'])){?>
            <a href="registrieren.php">Regsitrieren</a>
            <a href="login.php">Login</a>
        <?php } ?>
    </div>
</div>
</div>

```

2.1.1.3 Footer.php

Im Footer (Fussbereich der Website) wird ein Copyright-Symbol mit Start-Jahr und aktuellem Jahr angezeigt [37]. Der Footer wird wiederum mit der «style.css»-Datei formatiert, wobei das Aussehen an dasjenige des Headers angepasst wird. Hier ein Ausschnitt aus der entsprechenden Datei:

Code-Ausschnitt 3: footer.php

```

<div class="footer">
    <p>&copy; 2018 - <?php echo(date("Y")); ?> kantinet.ch | <a
    href="nutzungsbedingungen.php">Nutzungsbedingungen</a></p>
</div>

```

2.1.1.4 Functions.php

In der von mir als «functions.php» benannten Datei werden Funktionen und Eigenschaften definiert, die auf mehreren Unterseiten verwendet werden [32]. Die Datei wird auf allen Unterseiten eingebunden.

Einerseits wird die MySQL-Verbindung als Funktion geschrieben, sodass eine allfällige Änderung der Server-Zugangsdaten nur an einer Stelle vorgenommen werden muss:

Code-Ausschnitt 4: functions.php

```

//Diese Funktion stellt eine Verbindung zum MySQL-Server her. Die Datenbank-Login-Details
können im Nachhinein abgeändert werden.
function mysql_verbindung_1(){
    $con = mysqli_connect("localhost", "root", "", "kantinet");
    //Die Zeichencodierung wird auf UTF 8 gestellt.
    mysqli_query($con, "SET NAMES utf8");
    return $con;
}

```

Weiter wird in der Datei eine Session gestartet, weil Sessions auf fast allen Seiten genutzt werden und dafür zuerst ein Start initiiert werden muss.

Hauptsächlich werden in der «functions.php»-Datei allerdings Arrays beziehungsweise Auflistungen für die Sidebars und Dropdown-Menüs definiert. Die folgenden Arrays werden definiert: Kategorien der Bücherbörse, Liste mit Zürcher Kantonsschulen [31], Maturitätsprofile, mögliche Zustände der Bücher.

2.1.1.5 Sessions

Mit sogenannten Sessions, die mit PHP initiiert werden, hat man die Möglichkeit, Daten verschlüsselt im Browser zwischenzuspeichern. Dies hat den Vorteil, dass ebendiese Daten von allen Unterseiten bzw. Dateien der Website aufgerufen werden können. Im Rahmen dieser Arbeit verwende ich Sessions, um die ID des eingeloggten Benutzers im Browser zu

speichern. So kann jederzeit über alle PHP-Dateien überprüft werden, ob ein Benutzer eingeloggt ist und was dessen ID ist. So können auf den Benutzer zugeschnittene Informationen und Funktionen – wie Formulare, Buttons und Links – angezeigt werden.

Will man die Benutzer-ID in einer Variablen speichern, geht das folgendermassen [11]:

Code-Ausschnitt 5: Vorkommen in mehreren Dateien

```
$benutzer_id = $_SESSION['benutzer_id'];
```

Dies geht nur, wenn vorher die Session-Funktion gestartet wurde, was, wie im Abschnitt «functions.php» erklärt, bei allen Dateien automatisch gemacht wird.

2.1.1.6 Style.css

In der «style.css»-Datei werden, wie bereits erwähnt, die Style-Attribute der HTML-Elemente zentral in einer Datei festgelegt.

Der Footer wird in der «style.css» beispielsweise auf diese Art und Weise behandelt:

Code-Ausschnitt 6: style.css

```
.footer {
background: #D8D8D8;
height: 80px;
width: 100%;
margin-top: 20px;
float:left;
line-height: 1;
}

.footer a {
color : #151515;
text-decoration: none;
}

.footer a:hover {
color: #0431B4;
}

.footer p {
text-align: left;
font-size: 15px;
color: #151515;
padding-top: 32.5px;
padding-left: 20px;
}
```

Der Punkt bedeutet, dass ein HTML Element mit dem «class»-Attribut angesprochen wird. Der Name der CSS-Klasse folgt nach dem Punkt.

Es können ausserdem bestimmte Elemente, wie beispielsweise ein Link (a), innerhalb einer Klasse angesprochen werden. Über «.footer a» kann beispielsweise allen Links innerhalb der Klasse «footer» bestimmte Eigenschaften zugesprochen werden.

Die CSS-Datei wird in allen anderen Dateien eingebunden. Ruft der Client eine Seite auf, wird automatisch die «style.css»-Datei aufgerufen und die entsprechenden Elemente werden gemäss CSS-Code angepasst. Das Einbinden mit dem Link-Tag geschieht in HTML folgendermassen:

Code-Ausschnitt 7: Vorkommen in mehreren Dateien

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

2.1.2 Datenbankstruktur

Die Datenbank, die ich in Korrelation zu meiner Website aufsetze, ist in zwei Tabellen eingeteilt: Es gibt die «benutzer»- und die «buecher»-Tabelle. Man muss sich vorstellen,

dass man als Datenbankstruktur die Spaltentitel bestimmt. Neue Datenbankeinträge werden als Zeilen eingefügt, wobei jede Zeile für alle Spaltenüberschriften Daten enthält.

In den folgenden zwei Kapiteln wird die Datenbankstruktur (die Spaltenüberschriften) der beiden Tabellen aufgezeigt. Dabei wird nebst dem Namen der Spalte auch der jeweilige MySQL-Datentyp angegeben.

2.1.2.1 Die Tabelle «benutzer»

Name	Typ
<i>id</i>	<i>int(11)</i>
<i>datum_erstellt</i>	<i>datetime</i>
<i>datum_aktualisiert</i>	<i>datetime</i>
<i>datum_reaktiviert</i>	<i>datetime</i>
<i>datum_geloescht</i>	<i>datetime</i>
<i>deaktiviert</i>	<i>int(11)</i>
<i>vorname</i>	<i>tinytext</i>
<i>nachname</i>	<i>tinytext</i>
<i>email</i>	<i>tinytext</i>
<i>passwort</i>	<i>tinytext</i>
<i>benutzername</i>	<i>tinytext</i>
<i>stufe</i>	<i>int(11)</i>
<i>profil</i>	<i>tinytext</i>
<i>schule</i>	<i>tinytext</i>
<i>email_registrierung_kennung</i>	<i>int(11)</i>
<i>email_registrierung_ok</i>	<i>int(11)</i>
<i>email_reaktivierung_kennung</i>	<i>int(11)</i>

2.1.2.2 Die Tabelle «buecher»

Name	Typ
<i>id</i>	<i>int(11)</i>
<i>datum_erstellt</i>	<i>datetime</i>
<i>datum_aktualisiert</i>	<i>datetime</i>
<i>datum_verkauft</i>	<i>datetime</i>
<i>datum_verkauf_abgeschlossen</i>	<i>datetime</i>
<i>benutzer_id</i>	<i>int(11)</i>
<i>benutzer_geloescht</i>	<i>int(11)</i>
<i>Benutzer_deaktiviert</i>	<i>int(11)</i>
<i>gekauft_von_id</i>	<i>int(11)</i>
<i>schule</i>	<i>tinytext</i>
<i>isbn</i>	<i>tinytext</i>
<i>cover_link</i>	<i>tinytext</i>
<i>buch_titel</i>	<i>tinytext</i>
<i>autor</i>	<i>tinytext</i>
<i>jahr</i>	<i>int(11)</i>
<i>verlag</i>	<i>tinytext</i>
<i>kategorie</i>	<i>tinytext</i>
<i>zustand</i>	<i>tinytext</i>
<i>preis</i>	<i>tinytext</i>

2.2 Registrierungsverfahren

Die Registrierung erfolgt über die Datei «registrieren.php». Im Folgenden werden die wichtigsten Eigenschaften und Funktionen dieser Datei erläutert.

2.2.1 Registrierungsformular

Die in Abb. 4 ersichtlichen persönlichen Daten müssen zur Registrierung seitens des Benutzers angegeben werden. Klassenstufe, Profil und Kantonsschule werden mithilfe eines Drop-Down-Menüs ausgewählt. Die anderen Felder müssen vom Benutzer manuell über die Tastatur ausgefüllt werden. Das Formular wird mithilfe von HTML erstellt und in derselben Datei mithilfe von PHP ausgewertet. Dies geschieht über eine Schnittstelle von HTML und PHP: Die sogenannte POST-Übertragung. Diese Übertragung geschieht, wenn der Benutzer auf den «Registrieren»-Button klickt.



kantinet.ch Regsitrieren Login

Vorname

Nachname

Benutzername

Schul-E-Mail-Adresse

---Klassenstufe wählen---

---Profil wählen---

---Kantonsschule wählen---

Passwort

Registrieren

© 2018 - 2019 kantinet.ch | Nutzungsbedingungen

Abb. 4: Registrierungsformular (Eigenquelle: registrieren.php)

2.2.1.1 Drop-Down-Menüs des Formulars

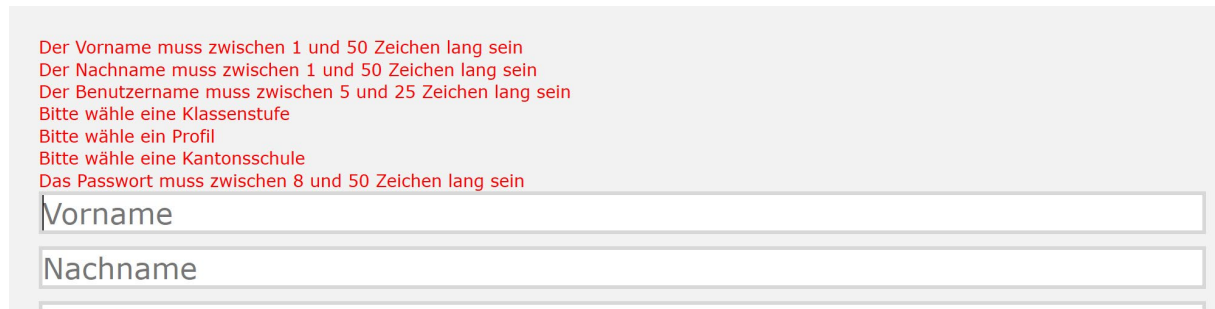
Die Drop-Down-Menüs des Formulars werden mithilfe der in der «functions.php»-Datei definierten Arrays erstellt:

Code-Ausschnitt 8: functions.php

```
<select name="profil">
  <option value="0">---Profil wählen---</option>
  <?php
    $anzahl_profile = count($profile_titel);
    for($i=0; $i<$anzahl_profile; $i++){?>
      <option value="<?php echo($profile[$i]) ?>"><?php echo($profile_titel[$i])
      ?></option>
    <?php } ?>
</select>
```

2.2.2 Bedingungen

Da man in die Formular-Felder theoretisch etwas Beliebiges von beliebiger Länge schreiben kann, muss der Inhalt mit diversen Bedingungen geprüft werden, bevor ein tatsächlicher Datenbankeintrag geschieht. Wenn etwas nicht den Vorgaben entspricht, bekommt der Benutzer eine oder mehrere Fehlermeldungen (siehe Abb. 5).



The screenshot shows a registration form with several input fields. Above the fields, there are several red error messages: "Der Vorname muss zwischen 1 und 50 Zeichen lang sein", "Der Nachname muss zwischen 1 und 50 Zeichen lang sein", "Der Benutzername muss zwischen 5 und 25 Zeichen lang sein", "Bitte wähle eine Klassenstufe", "Bitte wähle ein Profil", "Bitte wähle eine Kantonsschule", and "Das Passwort muss zwischen 8 und 50 Zeichen lang sein". Below these messages, there are input fields for "Vorname" and "Nachname".

Abb. 5: Registrierungsformular mit einigen Fehlermeldungen (Eigenquelle: registrieren.php)

Hier ein Beispiel einer Bedingung, die die Länge des Vornamens überprüft [3]:

Code-Ausschnitt 9: registrieren.php

```
if (strlen($vorname)>50 or strlen($vorname)<1 ){
    echo("<span class='statusmeldung'>Der Vorname muss zwischen 1 und 50 Zeichen lang
    sein</span></br>");
    $fehler=true;
}
```

Natürlich wird auch überprüft, ob Benutzername und E-Mail-Adresse bereits in der Datenbank vorhanden sind [4].

2.2.3 Überprüfung der E-Mail-Adresse

Eine eher kompliziertere Überprüfung ist diejenige der E-Mail-Adresse, weshalb sie hier explizit erklärt wird. Dabei müssen nicht nur die Länge und das Format [6], sondern auch die Domain überprüft werden. Dazu muss in mehreren Schritten aus der E-Mail-Adresse die Domain herausgelesen werden [5]:

Code-Ausschnitt 10: registrieren.php

```
//Trennung der E-Mail-Adresse am @ Zeichen und Speicherung der Teile in einem Array.
$email_array = explode('@', $email);
//Speicherung der gesamten Domain der E-Mail-Adresse.
$domainstring = $email_array[count($email_array)-1];
//Trennung der Domain an den "." Zeichen und Speicherung der Teile in einem Array
//-> Es könnte sich um eine Subdomain handeln
$domain_array = explode('.', $domainstring);
//Speicherung der Stamm-Domain in einer Variable -> Auswahl der letzten beiden Elemente des
Domain Arrays mit Punkt dazwischen.
$stamm_domain = $domain_array[count($domain_array)-2] . "." .
$domain_array[count($domain_array)-1];

//Kontrolle, ob die Domain erlaubt ist
if (!in_array($stamm_domain, $schulen_domains)){
    echo("<span class='statusmeldung'>Bitte gib deine persönliche Schul-E-Mail-Adresse an.
    Die E-Mail-Adresse muss zwischen 5 und 100 Zeichen lang sein</span></br>");
    $fehler=true;
}
```

2.2.4 Behandlung des Passworts

Das Passwort wird mit der Funktion «sha1» in einen sogenannten Hash umgewandelt. Es wird also nicht das eigentliche Passwort, sondern eine gehashte Form davon in der Datenbank gespeichert. Die Entschlüsselung ist praktisch unmöglich. Wenn sich der Benutzer später anmeldet, wird das von ihm eingegebene Passwort wiederum in einen

Hash umgewandelt und mit derjenigen gehashten Version des Passwortes in der Datenbank abgeglichen. Die Umwandlung in den Hash geschieht wie folgt [8]:

Code-Ausschnitt 11: registrieren.php

```
$password_hash = sha1($password);
```

2.2.5 Datenbankeintrag

Der Datenbankeintrag geschieht mittels SQL-Befehl, der in PHP mithilfe von «mysqli_query» übermittelt wird [9]:

Code-Ausschnitt 12: registrieren.php

```
$sql = "INSERT INTO benutzer (datum_erstellt, datum_aktualisiert, datum_reaktiviert,
datum_geloescht, deaktiviert, vorname, nachname, email, password, benutzername, stufe, profil,
schule, email_registrierung_kennung, email_registrierung_ok, email_reaktivierung_kennung)
VALUES ('$date_time_aktuell', '$date_time_aktuell', '$date_time_aktuell', '0000-00-00
00.00.00', '0', '$vorname', '$nachname', '$email', '$password_hash', '$benutzername',
'$stufe', '$profil', '$schule', '$kennung', '0', '0')";

mysqli_query($con, $sql);
```

2.2.6 E-Mail-Bestätigung und Aktivierung des Kontos

Zur Bestätigung der E-Mail-Adresse wird dem Benutzer eine automatisch generierte E-Mail gesendet. In dieser ist ein Link mit einer Zufallszahl als GET-Parameter vorhanden. GET ist wie POST eine Art der Datenübertragung zwischen zwei Dateien. Der GET-Parameter ist im Link ersichtlich und abänderbar, weshalb er für den hier beschriebenen Vorgang verwendet werden kann. Diese Zufallszahl wird für den späteren Abgleich zusätzlich in die Datenbank geschrieben (vgl. Kap. 2.2.5). Hier wird das Generieren der Zufallszahl und das Senden der E-Mail aufgezeigt:

Code-Ausschnitt 13: registrieren.php

```
//Generieren einer Zufallszahl zur E-Mail-Aktivierung des Benutzerkontos.
$kennung = rand(1,999999999);
//Generieren eines Bestätigungslinks mit der Zufallszahl als GET-Parameter im Link
$bestaetigungslink = $domain_url . "/aktivierung.php?kennung=" . $kennung;
//Senden des Bestätigungslinks.
mail($email, "Bestaetigung der E-Mail Adresse - kantinet", $bestaetigungslink);
```

Klickt der Benutzer auf den Bestätigungslink, den er per E-Mail bekommt, wird er zur Seite «aktivierung.php» weitergeleitet. Dort wird die Anfrage bearbeitet. Hier ein Ausschnitt des Files:

Code-Ausschnitt 14: aktivierung.php

```
$con = mysql_verbindung_1();
//Prüfen, ob die Bestätigungs-Kennung als Parameter im Link vorhanden ist.
//Sonst: Weiterleitung zur Startseite.
if(isset($_GET['kennung'])) {
    $kennung = $_GET['kennung'];
    //Wenn die Kennung im Link mit derjenigen in der Datenbank übereinstimmt, wird der
    Benutzer aktiviert -> Der entsprechende Parameter wird in der Datenbank abgeändert.
    //Sonst wird der Benutzer zur Startseite weitergeleitet.
    if (mysqli_num_rows(mysqli_query($con, "SELECT email_registrierung_kennung FROM
    benutzer WHERE email_registrierung_kennung='$kennung'"))==1) {
        mysqli_query($con, "UPDATE benutzer SET email_registrierung_ok = '1' WHERE
        email_registrierung_kennung='$kennung'");
        //Der Benutzer bekommt eine Bestätigung angezeigt und kann sich einloggen.
        echo('Dein Benutzerkonto wurde erfolgreich bestätigt. Du kannst dich jetzt auf
        der Website anmelden.');

??



```
Zum Login-Bereich<?php
 }
 else{
 header("Location: index.php");
 }
}
else{
 header("Location: index.php");
```


```

```
}
mysqli_close($con);
```

2.3 Login-Verfahren

Der Login-Prozess geschieht über die Datei «login.php». Wenn sich der Benutzer einloggen will, kann er dies entweder mit seinem Benutzernamen oder seiner E-Mail-Adresse tun. Wenn das Formular abgeschickt wird, werden die Daten, wie bei der Registrierung an dieselbe Datei zurückgesendet, wo sie dann verarbeitet werden.

The screenshot shows a web page for 'kantinet.ch'. In the top right corner, there are links for 'Regsitrieren' and 'Login'. The main content area contains a login form with two text input fields: the first is labeled 'Benutzername oder E-Mail-Adresse' and the second is labeled 'Passwort'. Below these fields is a 'Login' button. At the bottom of the page, there is a footer with the text '© 2018 - 2019 kantinet.ch | Nutzungsbedingungen'.

Abb. 6: Login-Formular (Eigenquelle: login.php)

2.3.1 Formular-Auswertung

Bei der Auswertung wird zunächst, analog zur Registrierung, geprüft, ob die Zeichenketten die erlaubte Länge haben oder nicht.

Im nächsten Schritt wird geprüft, ob der Benutzername oder die E-Mail-Adresse in der Datenbank vorhanden sind. Die Möglichkeit, dass Benutzername und E-Mail-Adresse identisch sind wird ebenfalls mit einbezogen.

Mithilfe des Benutzernamens oder der E-Mail-Adresse wird nun die ID des Benutzers ermittelt und damit der ganze Datensatz des Benutzers abgefragt. Im Folgenden der entsprechende Ausschnitt aus dem Code, wobei zuvor bei der Datenbank-Prüfung E-Mail-Adresse oder Benutzername in der Variable «\$benutzer» gespeichert wurde und den Variablen «\$benutzername_existiert» und «\$email_existiert» entsprechende Boolean-Werte zugewiesen wurden:

Code-Ausschnitt 15: login.php

```
if ($email_existiert==true and $benutzername_existiert==true){
    $res = mysqli_query($con, "SELECT id FROM benutzer WHERE benutzername='$benutzer'");
    $row = mysqli_fetch_array($res);
    $benutzer_id = $row[0];
}

elseif ($email_existiert==true and $benutzername_existiert==false){
    $res = mysqli_query($con, "SELECT id FROM benutzer WHERE email='$benutzer'");
    $row = mysqli_fetch_array($res);
    $benutzer_id = $row[0];
}

elseif ($email_existiert==false and $benutzername_existiert==true){
    $res = mysqli_query($con, "SELECT id FROM benutzer WHERE benutzername='$benutzer'");
    $row = mysqli_fetch_array($res);
    $benutzer_id = $row[0];
}
```


Das vom Benutzer eingegebene Passwort wird in einen Hash umgewandelt. Dieser wird anschliessend mit demjenigen in der Datenbank abgeglichen. Sofern das Konto nicht deaktiviert ist, wird der Benutzer nun eingeloggt.

2.4 Reaktivierungssystem

Bei jedem Login-Prozess wird eine Reaktivierungskontrolle durchgeführt. Dafür wird aus der «benutzer»-Tabelle der Datenbank der «datum-reaktiviert»-Wert abgefragt und es wird die Differenz zum aktuellen Datum gebildet [10]. In der Abb. 7 werden die entsprechenden Schritte des Login-Prozesses aufgeführt. Die Reaktivierungskontrolle wird ebenfalls über die «login.php»-Datei ausgeführt.

Das Reaktivierungssystem dient in erster Linie dazu, die Bücherbörse aktuell zu halten und Benutzerkonten von Personen, die die Kantonsschule schon seit längerer Zeit abgeschlossen haben, zu deaktivieren.

Ein deaktivierter Benutzer kann sich nicht mehr bei der Website anmelden. Des Weiteren werden seine Bücher von der Bücherbörse entfernt und können nicht mehr gekauft werden.

Reaktivierungskontrolle beim Login-Prozess

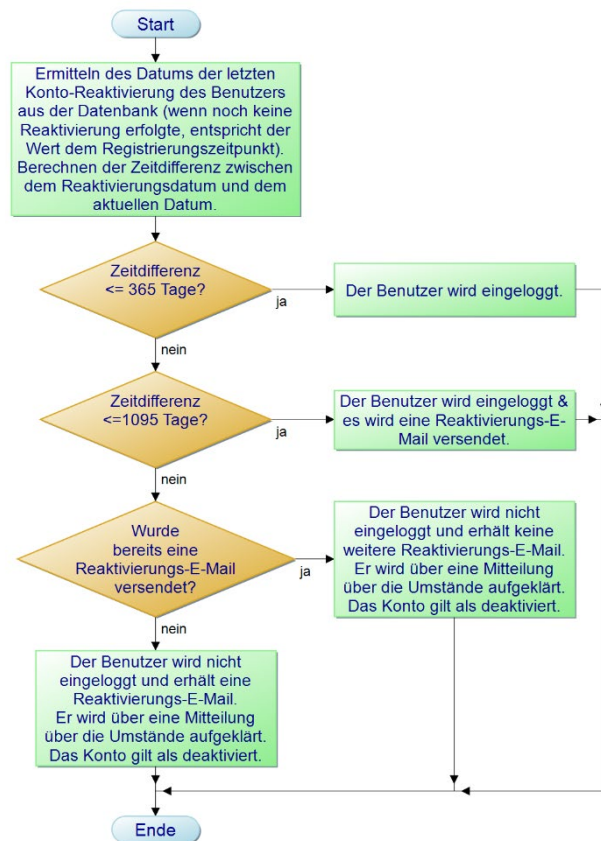


Abb. 7: Reaktivierungskontrolle beim Login-Prozess (Eigenquelle)

2.4.1 Reaktivierungsprozess

Wenn ein Benutzer eine Reaktivierungsmail bekommen hat, kann er, unabhängig davon, ob das Konto bereits deaktiviert wurde, sein Benutzerkonto über den per E-Mail erhaltenen Link reaktivieren. Dabei wird, analog zur Aktivierung nach der Registrierung, eine Zufallszahl als GET-Parameter in den Link integriert. Die Reaktivierung erfolgt über die Datei «reaktivierung.php». Hier ein Ausschnitt daraus

Code-Ausschnitt 16: reaktivierung.php

```

$con = mysql_verbindung_1();
//Prüfen, ob eine Kennung im Link vorhanden ist.
//Sonst: Weiterleitung zur Startseite.
if(isset($_GET['kennung'])){
    //Speichern der Kennung in einer Variablen.
    $kennung = $_GET['kennung'];
    //Wenn die Kennung stimmt, wird der folgende Code ausgeführt.
    //Sonst: Weiterleitung zur Startseite.
    if (mysqli_num_rows(mysqli_query($con, "SELECT email_reaktivierung_kennung FROM benutzer WHERE email_reaktivierung_kennung='$kennung'"))==1) {
        //Falls das Konto bereits gelöscht wurde, kann es nicht mehr reaktiviert werden
        $res = mysqli_query($con, "SELECT * FROM benutzer WHERE email_reaktivierung_kennung='$kennung'");
        $row = mysqli_fetch_assoc($res);
    }
}

```

```

$datum_geloescht = $row['datum_geloescht'];
$benutzer_id = $row['id'];
if($datum_geloescht != "0000-00-00 00:00:00"){
    echo("Das Konto wurde gelöscht und kann nicht mehr aktiviert werden");
}
else{
    //Speichern des aktuellen Datums in einer Variablen.
    $date_time_aktuell = date("Y-m-d H:i:s");
    //Entsprechende Anpassung der Datenbank zur Reaktivierung des
    Benutzerkontos.
    mysqli_query($con, "UPDATE benutzer SET deaktiviert = '0' WHERE
    id='$benutzer_id'");
    mysqli_query($con, "UPDATE benutzer SET datum_reaktiviert =
    '$date_time_aktuell' WHERE id='$benutzer_id'");
    mysqli_query($con, "UPDATE benutzer SET email_reaktivierung_kennung =
    '0' WHERE id='$benutzer_id'");
    mysqli_query($con, "UPDATE buecher SET benutzer_deaktiviert = '0' WHERE
    benutzer_id = '$benutzer_id'");
    echo('Dein Benutzerkonto wurde erfolgreich reaktiviert. Du kannst dich
    jetzt auf der Website anmelden.');
```

2.5 Bücherbörse

Die Bücher-Übersicht sowie die Buch-Registrierung werden über die Datei «buecherboerse.php» gesteuert. Für das Ausgeben der Details eines einzelnen Exemplars sowie für den Kaufprozess wurde die Datei «buch.php» eingerichtet.

2.5.1 Buch-Registrierung

2.5.1.1 ISBN-Nummer

Das zentrale Element bei der Registrierung eines neuen Buches auf der Website seitens des eingeloggtten Benutzers ist die ISBN-Nummer. Es handelt sich dabei um die sogenannte Internationale Standardbuchnummer, welche hauptsächlich der eindeutigen Kennzeichnung von Büchern dient [44]. Diese 10- oder 13-stellige Nummer ist bei den meisten Büchern auf der Rückseite oberhalb des Barcodes ersichtlich.

Der Verwendungszweck der ISBN-Nummer auf der Website ist der Folgende: Über das in Abb. 8 ersichtliche Formular, welches sich oberhalb der Bücher-Übersicht im Untermenü «Bücherbörse» befindet, kann ein neues Buch registriert werden, Dabei genügt es vorerst, falls vorhanden, nur die ISBN-Nummer des zu registrierenden Exemplars einzugeben. Damit können gewisse Buchinformationen automatisch von einer externen Plattform zugezogen werden, sodass der Benutzer diese im nächsten Schritt nicht mehr eingeben muss.

Wenn keine ISBN-Nummer vorhanden ist, gelangt der Benutzer über den Link «Keine ISBN?» direkt zum eigentlichen Registrierungsformular. Dieses muss allerdings manuell ausgefüllt werden. Ausserdem wird das Standard-Cover verwendet.

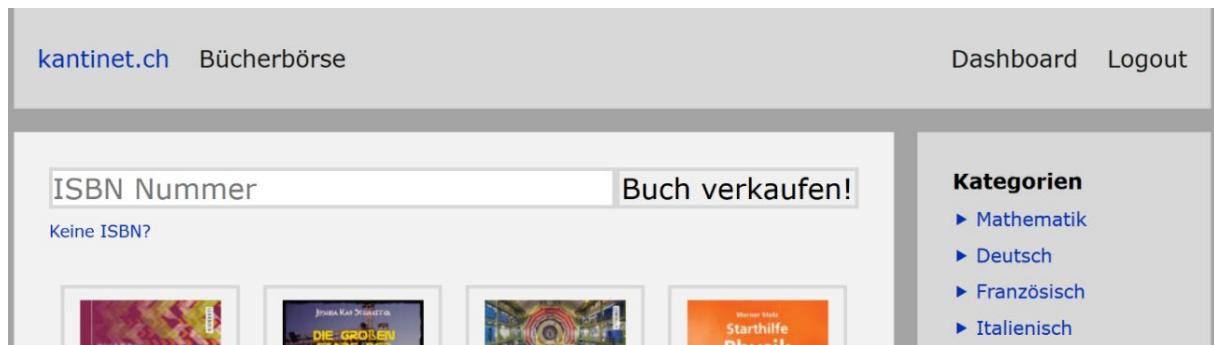


Abb. 8: Buch-Registrierung mithilfe der ISBN-Nummer (Eigenquelle: buecherboerse.php) (sichtbare Cover-Bilder: [50])

2.5.1.2 Google Books API

Bei der zuvor erwähnten externen Plattform handelt es sich um Google Books. Konkret ist dabei von einer sogenannten API, also einer Schnittstelle, die Rede. Gibt der Benutzer eine ISBN-Nummer ein und schickt das Formular ab, so wird eine Anfrage an Google Books gesendet [15]. Dabei wird die ISBN-Nummer in der Variablen «\$isbn» gespeichert. Hier ein Ausschnitt aus der Datei «buecherboerse.php»:

Code-Ausschnitt 17: buecherboerse.php

```
//Eine ISBN-Nummer ist immer eine 10- oder 13-stellige numerische Zeichenkette.
if(!is_numeric($isbn) or (strlen($isbn)!=10 and strlen($isbn)!=13)){
    echo("<span class='statusmeldung'>Die ISBN-Nummer ist ungültig. Eine ISBN-Nummer ist
    eine 10- oder 13-stellige Zahl.</span></br>");
    $isbn_ok=false;
}

else{
    //Die vom Benutzer eingegebene ISBN-Nummer wird nun an die Google Books API gesendet.
    //Das @ Zeichen verhindert eine Fehlermeldung falls die Google Seite nicht existiert.
    @$str = file_get_contents("https://www.googleapis.com/books/v1/volumes?q=isbn:$isbn");

    //Die von Google zurückgegebene Seite ist eine JSON-Datei mit den entsprechenden
    gesuchten Informationen zum Buch.
    //Die JSON-Datei wird mit der folgenden Funktion in einen assoziativen Array
    umgewandelt, damit mit den Informationen in PHP gearbeitet werden kann.
    $json = json_decode($str, true);

    //Die folgenden Bedingungen kontrollieren, ob von den jeweiligen gesuchten Buch-
    Eigenschaften Informationen in der Google Books API vorhanden sind.
    //Falls entsprechende Buch-Informationen gefunden werden können, werden diese in den am
    Anfang dieser Datei als leer definierten Variablen gespeichert.

    if(isset($json['items']['0']['volumeInfo']['imageLinks']['thumbnail'])){
        $cover_link = $json['items']['0']['volumeInfo']['imageLinks']['thumbnail'];
    }

    if(isset($json['items']['0']['volumeInfo']['title'])){
        $titel = $json['items']['0']['volumeInfo']['title'];
    }

    if(isset($json['items']['0']['volumeInfo']['authors']['0'])){
        $autor = $json['items']['0']['volumeInfo']['authors']['0'];
    }

    if(isset($json['items']['0']['volumeInfo']['publisher'])){
        $verlag = $json['items']['0']['volumeInfo']['publisher'];
    }

    if(isset($json['items']['0']['volumeInfo']['publishedDate'])){
        //Hier wird mithilfe einer Funktion das Jahr aus dem Veröffentlichungsdatum
        "herausgeschnitten"
        $jahr = substr($json['items']['0']['volumeInfo']['publishedDate'], 0, 4);
    }
}
```

2.5.1.3 Buch-Registrierungsformular

Wenn im vorherigen Schritt entsprechende Informationen zu den jeweiligen Buch-Eigenschaften gefunden wurden, werden diese gespeichert und an das eigentliche Buch-Registrierungsformular gesendet. Dieses wird ebenfalls über die Datei «buecherboerse.php» gesteuert. Die Informationen werden mithilfe von «POST» an dieselbe Datei zurückgesandt.

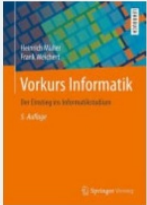
Wenn ein Cover-Bild in der Datenbank von Google als Link hinterlegt ist, wird dieses als Cover für Bücherbörse zugezogen (siehe Abb. 9). Dabei wird die Bild-Datei nicht auf dem eigenen Server gespeichert, sondern bei jedem Ladevorgang von Google angefragt. Dies spart Speicherplatz. Wenn kein Bild gefunden wurde, wird das Standard-Cover verwendet (siehe Abb. 10). Der Nachteil ist, dass eine ständige Verbindung zu Google notwendig ist und eine Abhängigkeit besteht. Wird das Bild später nicht mehr gefunden, wird ebenfalls das Standard-Cover verwendet [18].

Das Beispiel in Abb. 9 zeigt einen Fall, bei dem mithilfe der ISBN-Nummer alle angefragten Informationen zum Buch gefunden wurden. Angefragt werden Cover-Link, Buch-Titel, Autor, Verlag und Veröffentlichungsjahr. Kategorie, Zustand und Preis müssen immer manuell vom Benutzer eingegeben beziehungsweise ausgewählt werden.

Wenn keine oder nicht alle der fünf angefragten Informations-Daten gefunden werden, muss der Benutzer auch diese manuell eingeben. Das Beispiel in Abb. 10 zeigt einen Fall, bei dem gar keine Informationen gefunden wurden oder bei dem der Benutzer die Option «Keine ISBN-Nummer vorhanden» gewählt hat.

kantinet.ch Bücherbörse

Das



Vorkurs Informatik

Heinrich Müller

Springer Vieweg

2017

---Kategorie wählen---

---Zustand des Artikels angeben---

Preis in CHF


Buch verkaufen

Kat

- ▶ M
- ▶ D
- ▶ Fi
- ▶ It
- ▶ E
- ▶ S
- ▶ R
- ▶ C
- ▶ L
- ▶ P
- ▶ C
- ▶ B
- ▶ G

Abb. 9: Buch-Registrierungsformular mit Informationen aus der Google Books API (Eigenquelle: buecherboerse.php) (sichtbares Cover-Bild: [50])

kantinet.ch
Bücherbörse
Da



---Kategorie wählen---

---Zustand des Artikels angeben---

Buch verkaufen

Ka

► M

► C

► F

► I

► E

► S

► F

► C

► L

► P

► C

► E

► C

Abb. 10: Leeres Buch-Registrierungsformular (Eigenquelle: buecherboerse.php) (sichtbares Cover-Bild: [48])

2.5.1.4 Auswertung des Buch-Registrierungsformulars

Auch die Eingaben dieses Formulars werden zuerst mithilfe diverser Bedingungen auf korrekte Länge und Form überprüft. Eine Besonderheit ist die Behandlung des eingegebenen Verkaufspreises. Dieser wird unabhängig davon, wie viele Nachkommastellen eingegeben wurden, auf fünf Rappen genau gerundet. Nach dem Vorgang ist der Preis demnach eine Zahl mit zwei Nachkommastellen. Das Runden funktioniert wie folgt [12] [13] [14]:

Code-Ausschnitt 18: buecherboerse.php

```
//Hier wird der vom Benutzer eingegebene Preis zuerst in einen Float umgewandelt und
anschliessend auf fünf Rappen genau gerundet.
$preis_float = floatval($preis);
$preis_float_gerundet = number_format(round(($preis_float + 0.000001) * 20) / 20, 2, '.', '');
//Hier wird ein Maximalpreis gesetzt. Mehr als 100 Fr. sind unrealistisch für ein Buch.
if ($preis_float_gerundet > 100){
    echo("<span class='statusmeldung'>Der Preis darf nicht mehr als 100 Fr.
    betragen</span></br>");
    $fehler=true;
}
```

Zum Schluss wird wiederum ein entsprechender Datenbankeintrag gemacht. Dabei wird der Tabelle «buecher» ein Datensatz angefügt:

Code-Ausschnitt 19: buecherboerse.php

```
if ($fehler==false){
    echo("<span class='statusmeldung'>Vielen Dank für die Registrierung des
    Buches!</span>");

    $date_time_aktuell = date("Y-m-d H:i:s");

    $res1 = mysqli_query($con, "SELECT schule FROM benutzer WHERE id='$benutzer_id'");
    $row1 = mysqli_fetch_array($res1);
    $benutzer_schule = $row1[0];

    $res = "INSERT INTO buecher (datum_erstellt, datum_aktualisiert, datum_verkauft,
    datum_verkauf_abgeschlossen, benutzer_id, benutzer_geloescht, benutzer_deaktiviert,
```

```

gekauft_von_id, schule, isbn, cover_link, buch_titel, autor, jahr, verlag, kategorie,
zustand, preis) VALUES ('$date_time_aktuell', '$date_time_aktuell', '0000-00-00
00.00.00', '0000-00-00 00.00.00', '$benutzer_id', '0', '0', '0', '$benutzer_schule',
'$isbn', '$cover_link', '$titel', '$autor', '$jahr', '$verlag', '$kategorie',
'$zustand', '$preis_float_gerundet');"

mysqli_query($con, $res);
}

```

2.5.2 Bücher-Übersicht

Das zentrale Element der Website ist die Bücher-Übersicht. Im Untermenü «Bücherbörse» können von eingeloggten Benutzern alle registrierten Bücher eingesehen werden.

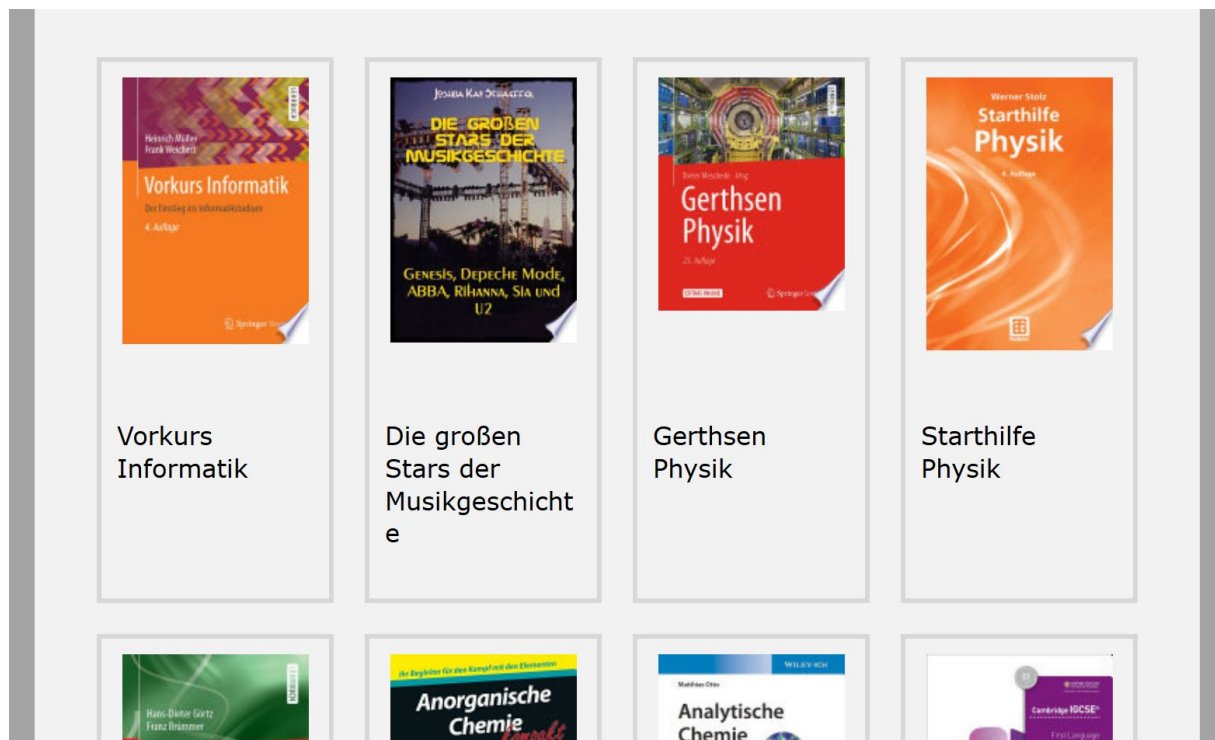


Abb. 11: Elemente aus der Bücher-Übersicht (Eigenquelle: *buecherboerse.php*) (sichtbare Cover-Bilder: [50])

2.5.2.1 Filtermöglichkeiten

2.5.2.1.1 Filter-Sidebar

Rechts neben der Bücher-Übersicht ist eine Sidebar platziert. Eine solche Seitenleiste verkörpert sozusagen den Seitenbereich einer Website. Dort werden ähnlich wie in der Kopfzeile Navigations-Links angezeigt.

Die Sidebar der Bücherbörse wird nur zusammen mit der Bücher-Übersicht angezeigt. Sie bietet die Möglichkeit, die Auswahl anzuzeigender Bücher einzugrenzen und ist die Steuereinheit des Filters. Es kann nach Kategorie und/oder nach Kantonsschule des Verkäufers gefiltert werden.

Wenn ein Eintrag der Sidebar angewählt beziehungsweise ein Filter angewendet wurde, wird der entsprechende ausgewählte Eintrag der Sidebar in der Formatierung hervorgehoben. Es kann jeweils nur eine oder gar keine Kategorie beziehungsweise Schule ausgewählt werden. In Abb. 12 ist eine Situation ersichtlich, in der «Mathematik» als Kategorie ausgewählt wurde. Dies ist an der schwarzen Farbe und der fetten Hervorhebung erkennbar. Neben dem Sidebar-Titel «Kategorien» ist ausserdem ein Haken [27] platziert, welcher zeigt, dass eine Kategorie ausgewählt wurde. Wird eine andere Kategorie ausgewählt, wird «Mathematik» wieder in normaler, blauer Formatierung angezeigt. Der

Kategorie-Filter kann in diesem Fall durch erneutes Klicken auf «Mathematik» entfernt werden, wodurch auch besagter Haken entfernt wird.

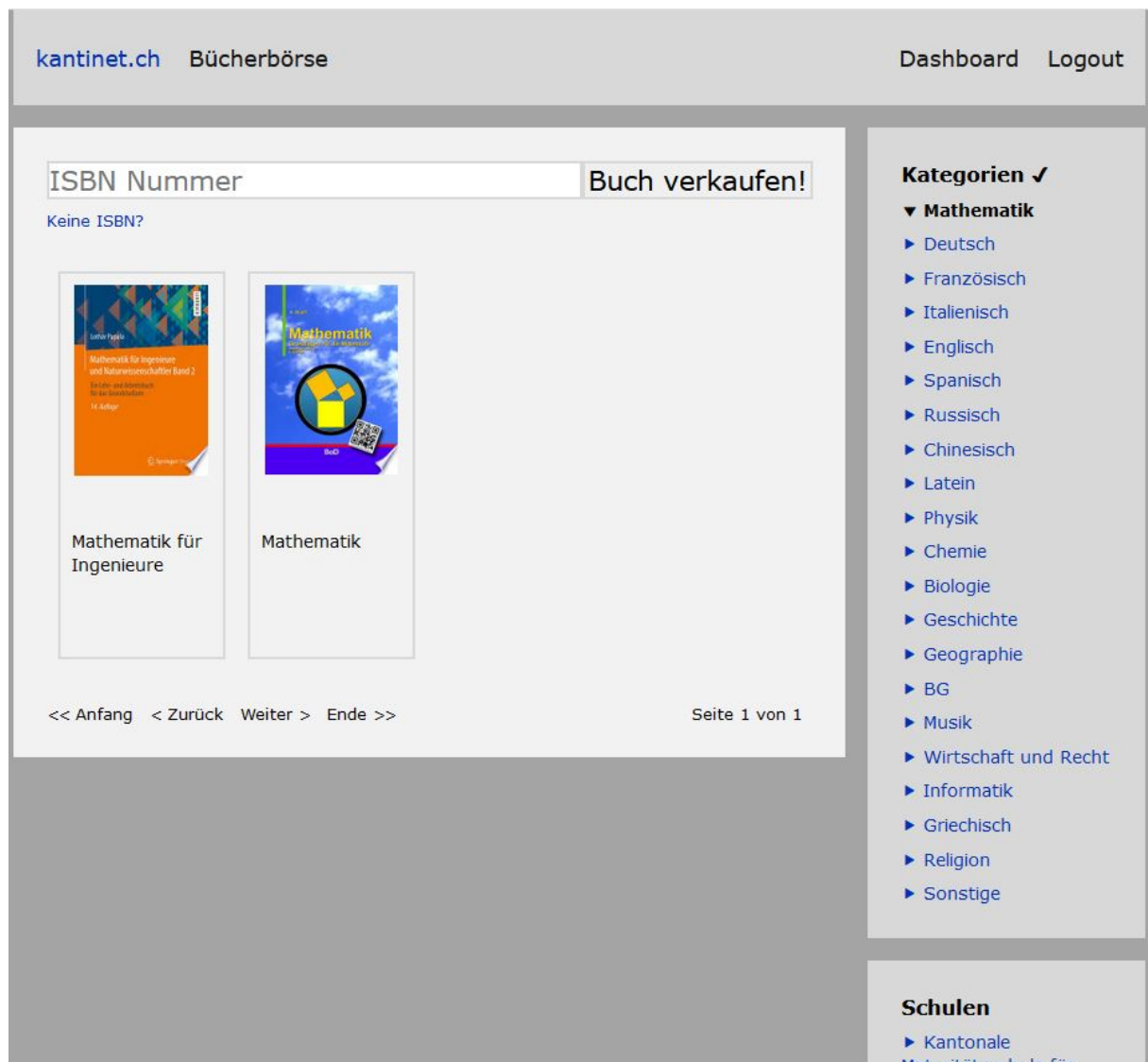


Abb. 12: Bücher-Übersicht mit einer ausgewählten Kategorie und keiner ausgewählten Schule (Eigenquelle: buecherboerse.php) (sichtbare Cover-Bilder: [50])

Die Auswahl an Kategorien und Schulen ist, wie in einem vorherigen Kapitel erklärt, in der Datei «dashboard.php» definiert. Diese wird in jede andere Datei miteinbezogen, weshalb mit den dort aufgeführten Arrays auch hier gearbeitet werden kann.

Die Einträge der Sidebar werden mithilfe einer For-Schleife aus den Arrays ausgelesen. Bei ausgewählter Kategorie und/oder Schule wird mithilfe eines GET-Parameters ein entsprechender Eintrag im Link hinterlegt. So kann in der Schleife jeweils auch ermittelt werden, ob eine Kategorie oder Schule ausgewählt wurde.

Hier ein Ausschnitt des Codes, welcher das Kategorie-Sidebar-Element steuert (Die Funktionsweise der Sidebar-Einheit für die Auswahl der Kantonsschule funktioniert analog):

Code-Ausschnitt 20: buecherboerse.php

```
for ($i=0; $i < count($kategorien); $i++){
    //Falls die ausgewählte Kategorie in der Schleife "an die Reihe kommt", wird dafür
    gesorgt, dass diese eine andere Farbe hat und fett hervorgehoben wird.
```

```

//Bei erneutem Klicken auf die ausgewählte Kategorie wird der Parameter in der URL
gleich null gesetzt. Dies ist gleichwertig, wie wenn gar keine Kategorie in der URL
vorhanden ist.
//Das @ verhindert eine Fehlermeldung, falls kein "$_GET['kategorie']"-Wert existiert.
if(@$_GET['kategorie']==$kategorien[$i]){
    $neue_url_kategorie = str_replace("kategorie=".$kategorien[$i], "kategorie=0",
    $aktuelle_url);

    if(isset($_GET['p'])){
        //Falls ein "p" Wert in der URL vorhanden ist wird dieser auf 1 gesetzt,
        damit man beim erneuten Klicken auf die ausgewählte Kategorie wieder auf
        die erste Seite gelangt.
        $neue_url_kategorie = str_replace("p=".$_GET['p'], "p=1",
        $neue_url_kategorie);
    }

    ?><a style="color:#000; font-weight: bold;" href="<?php
    echo($neue_url_kategorie); ?>"&#11206; <?php echo($kategorien_titel[$i]);
    ?></a><?php
}
else {
    if(!isset($_GET['schule'])){
        ?><a href="?kategorie=<?php echo($kategorien[$i]); ?>"&#11208; <?php
        echo($kategorien_titel[$i]); ?></a><?php
    }
    else{
        //Hier wird dafür gesorgt, dass ein allfälliger Parameter zur Schule
        nicht verloren geht und in der neuen URL immer noch vorhanden ist.
        ?><a href="?kategorie=<?php echo($kategorien[$i]); ?>&schule=<?php
        echo($_GET['schule']); ?>"&#11208; <?php echo($kategorien_titel[$i]);
        ?></a><?php
    }
}
}
}

```

2.5.2.1.2 Hintergrund-Technik des Filters

In diesem Kapitel wird aufgezeigt, wie die Datenbankabfrage der Bücher für die Bücher-Übersicht mit Einbezug allfällig angewandter Filter funktioniert. Dabei spielen SQL-Befehle eine zentrale Rolle. Mit ihnen kann die Datenbankabfrage nach gewissen Kriterien eingegrenzt werden.

In diesem Kapitel wird insofern vorgegriffen, als dass bereits Elemente der später genauer geschilderten Blätterfunktion aufgezeigt werden. Diese sind allerdings auch für die hier erklärte Datenbankabfrage nötig.

Im Folgenden der entsprechende Ausschnitt aus dem Code [19] [20] [21]:

Code-Ausschnitt 21: buecherboerse.php

```

//Hier werden die Standard-Bedingungen für die spätere Datenbank-Abfrage definiert. Es werden
nur Bücher angezeigt, die noch nicht verkauft wurden und nicht von einem deaktivierten oder
gelöschten Benutzer stammen.
$bedingung_standard = "WHERE gekauft_von_id = '0' AND benutzer_geloescht = '0' AND
benutzer_deaktiviert = '0'";
$bedingung_schule = "";
$bedingung_kategorie = "";

//Falls eine Kategorie und/oder eine Schule in der Sidebar ausgewählt wurde, wird im Folgenden
eine entsprechende Bedingung aufgestellt, die anschliessend der Standard-SQL-Anfrage angefügt
wird.
//Falls in der URL Parameter vorhanden sind, wird geprüft, ob die entsprechende Kategorie oder
die Schule auch wirklich im in der Datei "functions.php" definierten Array vorhanden sind.

if(isset($_GET['kategorie'])){
    if(in_array($_GET['kategorie'], $kategorien)==true){
        $bedingung_kategorie = "AND kategorie = '" . $_GET['kategorie'] . "'";
    }
}

if(isset($_GET['schule'])){
    if(in_array($_GET['schule'], $schulen)==true){
        $bedingung_schule = "AND schule = '" . $_GET['schule'] . "'";
    }
}

```



```
//Nun werden die Bedingungen in der SQL-Sprache entsprechend zusammengefügt.
$bedingung_gesamt = $bedingung_standard . $bedingung_kategorie . $bedingung_schule;

//Hier wird eine MySQL-Anfrage erstellt, die die oben definierten Bedingungen berücksichtigt
und nur dazu dient, die gesamte Anzahl Datensätze zu bestimmen.
$res_gesamt = mysqli_query($con, "SELECT * FROM buecher $bedingung_gesamt");

//Für die spätere Blätterfunktion wird die Anzahl ausgewählter Datensätze bestimmt.
//Daraus wird die Anzahl Seiten bestimmt, welche auch für die Datenbankabfrage benötigt wird.
$anz_datensaetze = mysqli_num_rows($res_gesamt);
$anz_datensaetze_pro_seite = 12;
$anz_seiten = ceil($anz_datensaetze / $anz_datensaetze_pro_seite);

//Die $p Variable bestimmt die aktuell angezeigte Seite, der Standardwert von $p ist 1
if(isset($_GET['p'])){
    $p=intval($_GET['p']);
    //Sicherstellen, dass die im Link hinterlegte Seite existiert.
    if($p <= 0 or $p > $anz_seiten){
        $p=1;
    }
}

else{
    $p=1;
}

if($anz_datensaetze==0){
    echo("Leider sind in dieser Kategorie keine Bücher vorhanden.");
}

//Mit den folgenden Befehlen wird das entsprechende "Intervall" an Datensätzen ausgewählt. Auf
der ersten Seite werden beispielsweise die Datensätze 0-11 angezeigt, also die ersten 12
Datensätze.
//Das Intervall wird im SQL Befehl mit dem Parameter "LIMIT" definiert.
$limit_start = ($anz_datensaetze_pro_seite*$p) - $anz_datensaetze_pro_seite;

$res = mysqli_query($con, "SELECT id, cover_link, buch_titel FROM buecher $bedingung_gesamt
ORDER BY id DESC LIMIT $limit_start,$anz_datensaetze_pro_seite");
```

2.5.2.2 Buchrahmen

Die Bücher-Übersicht setzt sich im Wesentlichen aus den innerhalb von Rahmen dargestellten Büchern zusammen (siehe Abb. 11). Dabei wird innerhalb der einzelnen Rahmen neben dem Buch-Cover der Titel des Exemplars angezeigt. Wenn innerhalb eines Titels ein Wort länger als die festgelegte Breite des Rahmens ist, wird das Wort auf mehrere Zeilen verteilt [26]. Die Rahmen werden mithilfe einer Schleife erzeugt. Dabei wird die zuvor definierte Datenbankabfrage (\$res) als Grundlage zugezogen. Diese beinhaltet bereits die auf die aktuelle Seitenzahl und allfällige Filter zugeschnittene Auswahl an Büchern. Pro Seite werden, wie im Code des letzten Kapitels ersichtlich, 12 Rahmen beziehungsweise Datenbankeinträge angezeigt. Diese werden auf 3 Zeilen aufgeteilt. Hier wird aufgezeigt, wie die erwähnte Schleife funktioniert [22] [23] [24] [25]:

Code-Ausschnitt 22: buecherboerse.php

```
//Die While-Schleife gibt die Datensätze in Form von Buch-Rahmen mit Titel und Cover aus.
//Beim Klicken auf einen der Buch-Rahmen wird man zur entsprechenden Buch-Informations-Seite
weitergeleitet.

while($row = mysqli_fetch_object($res)){ ?>
    <a href="buch.php?id=?php echo $row->id ?>">
        <div class="buch-rahmen">
            <div class="buch-rahmen-bild">
                
            </div>

            <div class="buch-rahmen-text">
                <?php echo $row->buch_titel; ?>
            </div>

        </div>
    </a>
```

```
<?php }
```

Ein grosser Teil der Programmierung der Bücher-Ansicht, bezogen darauf, was der Benutzer auch wirklich sehen kann, spielt sich in der Datei «style.css» ab. Deshalb folgt hier ein kleiner Ausschnitt daraus:

Code-Ausschnitt 23: style.css

```
.buch-rahmen{
    width: 146.5px;
    height: 345px;
    border: 3px solid #D8D8D8;
    border-radius: 0px;
    float: left;
    margin-right: 10px;
    margin-left: 10px;
    margin-bottom: 10px;
    margin-top: 10px;
}
```

2.5.2.3 Blättermöglichkeit

Ohne eine Begrenzung der Anzahl anzuzeigender Datensätze pro Seite, könnte die Seite theoretisch mit zunehmender Anzahl registrierter Bücher immer länger werden. Dies würde zu allgemeiner Unübersichtlichkeit und langen Seiten-Ladezeiten führen.

Deshalb werden pro Seite, wie zuvor erwähnt, jeweils nur 12 Datensätze angezeigt. Sortiert wird nach Registrierungsdatum wobei das neuste Buch zuerst angezeigt wird. Auf der ersten Seite sind demnach die neusten, auf der letzten die ältesten Exemplare ersichtlich.

Das Aufteilen in Unterseiten bedingt einerseits einer neuen Steuereinheit (siehe Abb. 13) um das «Blättern» zu ermöglichen, andererseits eines entsprechenden Parameters im Link («\$_GET['p']»), welcher die aktuelle Seitenzahl speichert. Hier der betreffende Code-Ausschnitt [16] [17] [28]:

<< Anfang < Zurück Weiter > Ende >>

Seite 1 von 2

Abb. 13: Blättermöglichkeit im Fussbereich der Bücher-Anzeige (Eigenquelle: buecherboerse.php)

Code-Ausschnitt 24: buecherboerse.php

```
//Im Folgenden wird die Blätterfunktion ausgeführt.
//Die aktuelle URL (der Stammordnerpfad) wird in den nächsten Schritten benötigt
$aktuelle_url = $_SERVER['REQUEST_URI'];

//Es könnte sein, dass der "p"-Wert in der URL ($_GET['p']) nicht mit der "$p"-Variable
übereinstimmt, da jemand die URL manuell bearbeitet hat.
//Falls die "$p"-Variable nicht 1 ist, stimmt der Wert in der URL sicher damit überein und es
existiert sicher ein "p"-Wert in der URL.
//Falls eine Kategorie und/oder eine Schule im Link gespeichert sind dürfen diese beim
Blättern nicht verloren gehen.
//Daher wird die aktuelle URL im "p"-Parameter durch ein String-replacement bearbeitet und der
neue "p"-Wert wird gesetzt.
if($p != 1){ ?>
    <span class="blaetter-link"><a href="<?php echo(str_replace("p=".$p, "p=1",
    $aktuelle_url)); ?>"><< Anfang</a></span>
    <span class="blaetter-link"><a href="<?php echo(str_replace("p=".$p, "p=" . ($p-1),
    $aktuelle_url)); ?>">< Zurück</a></span>
<?php }

//Auf der ersten Seite können die Links "Zurück" und "Anfang" nicht angeklickt werden.
else{ ?>
    <span class="blaetter-link"><< Anfang</span>
    <span class="blaetter-link">< Zurück</span>
<?php }

//Hier könnte $p gleich 1 sein und der "p"-Wert in der URL könnte nicht damit übereinstimmen.
//Wenn "$p" gleich 1 ist, könnte auch gar kein "p"-Wert in der URL gespeichert sein.
//$p könnte mit dem "p"-Wert in der URL übereinstimmen.
```

```

if($p<$anz_seiten){
if($p=1){
//Kontrolle ob in der URL ein "p"-Wert gespeichert ist, wenn ja wird er überschrieben
if(isset($_GET['p'])){ ?>
    <span class="blaetter-link"><a href="<?php echo(str_replace("p=".$_GET['p'],
    "p=" . ($p+1), $aktuelle_url)); ?>">Weiter </a></span>
    <span class="blaetter-link"><a href="<?php echo(str_replace("p=".$_GET['p'],
    "p=" . $anz_seiten, $aktuelle_url)); ?>">Ende </a></span>
<?php }

//Falls kein "p"-Wert in der URL gespeichert ist, wird einer angefügt.
//Hier spielt es zusätzlich eine Rolle, ob bereits andere GET-Parameter in der URL
vorhanden sind ("&"- oder "?"-Zeichen).
else{
    if(isset($_GET['kategorie']) or isset($_GET['schule'])){ ?>
        <span class="blaetter-link"><a href="<?php echo($aktuelle_url . "&p=" .
        ($p+1)); ?>">Weiter </a></span>
        <span class="blaetter-link"><a href="<?php echo($aktuelle_url . "&p=" .
        $anz_seiten); ?>">Ende </a></span>
        <?php }

    else{ ?>
        <span class="blaetter-link"><a href="<?php echo($aktuelle_url . "?p=" .
        ($p+1)); ?>">Weiter </a></span>
        <span class="blaetter-link"><a href="<?php echo($aktuelle_url . "?p=" .
        $anz_seiten); ?>">Ende </a></span>
        <?php }
    }
}

//Falls die "$p"-Variable nicht 1 ist, stimmt der Wert in der URL sicher damit überein und es
existiert sicher ein "p"-Wert in der URL. Dieser kann einfach überschrieben werden.
else{ ?>
    <span class="blaetter-link"><a href="<?php echo(str_replace("p=".$_p, "p=" . ($p+1),
    $aktuelle_url)); ?>">Weiter </a></span>
    <span class="blaetter-link"><a href="<?php echo(str_replace("p=".$_p, "p=" .
    $anz_seiten, $aktuelle_url)); ?>">Ende </a></span>
    <?php }
}
//Auf der letzten Seite können die Links "Weiter" und "Ende" nicht angeklickt werden.

else{ ?>
    <span class="blaetter-link">Weiter </span>
    <span class="blaetter-link">Ende </span>
<?php }

//Hier werden die aktuelle Seite und die gesamte Anzahl Seiten ausgegeben (nur wenn mehr als 0
Datensätze existieren).
if($anz_datensaetze>0){ ?>
    <span class="blaetter-link" style="float:right;">Seite <?php echo($p); ?> von <?php
    echo ($anz_seiten); ?></span>
<?php }

```

2.5.3 Buchdetails-Ansicht

Klickt der Benutzer auf ein Exemplar der Bücher-Übersicht, wird er an die Datei «buch.php» weitergeleitet. Dabei wird die id des Buches mithilfe eines GET-Parameters mitgesendet.

Die Buch-Informationen werden mittels einer HTML-Tabelle [35] dargestellt. Das Buch-Cover wird links oben angezeigt. Durch einen Klick auf den «Kaufen»-Button [36] kann das Buch gekauft werden (siehe Abb. 14).



Titel	Vorkurs Informatik
Autor	Heinrich Müller
Jahr	2015
Verlag	Springer-Verlag
Kategorie	Informatik
ISBN-Nummer	3658081023
Zustand	Neu
Kantonsschule	Kantonsschule Limmattal
Einstellungsdatum	19.08.2019
Verkäufer	marc.meiners
Preis	16.00 Fr.

Abb. 14: Buchdetails-Ansicht (Eigenquelle: buch.php) (sichtbares Cover-Bild: [50])

Mit dem folgenden Code werden die anzuzeigenden Buch-Informationen aus der Datenbank abgefragt und in Variablen gespeichert [33]:

Code-Ausschnitt 25: buch.php

```
//Die Buch-Informationen werden aus der Datenbank abgefragt und in Variablen gespeichert.
$res = mysqli_query($con, "SELECT * FROM buecher WHERE id='$buch_id'");
//Die id des angemeldeten Benutzers wird ebenfalls in einer Variablen gespeichert, im Falle
eines Kaufes seitens des angemeldeten Benutzers wird diese benötigt.
$kaeuer_id = $_SESSION['benutzer_id'];
$row = mysqli_fetch_assoc($res);
$buch_titel = $row['buch_titel'];
$autor = $row['autor'];
$jahr = $row['jahr'];
$verlag = $row['verlag'];
//Einige Werte sind in Kurzform in der Datenbank gespeichert -> Der "Anzeigename" dieser Werte
ist in der "functions.php" in einem separaten Array an derselben Position im Array
gespeichert.
$zustand = $zustaende_titel[array_search($row['zustand'], $zustaende)];
$kategorie = $kategorien_titel[array_search($row['kategorie'], $kategorien)];
$isbn = $row['isbn'];
$verkaeuer_id = $row['benutzer_id'];

//Um den Benutzernamen des Verkäufers herauszufinden, ist eine erneute Datenbankabfrage,
dieses Mal aus der "benutzer"-Tabelle, von Nöten.
$row1 = mysqli_fetch_array(mysqli_query($con, "SELECT benutzername FROM benutzer WHERE
id='$verkaeuer_id'"));
$verkaeuer_benutzername = $row1[0];

$schule_verkaeuer = $schulen_titel[array_search($row['schule'], $schulen)];

$datum_einstellung = date('d.m.Y', strtotime($row['datum_erstellt']));
$preis = $row['preis'];
```

2.5.4 Kaufprozess

Wenn sich ein Benutzer zum Kauf eines Buches entscheidet, kann er den «Kaufen»-Button betätigen. Wenn er dies tut werden einerseits entsprechende Bearbeitungen der Datenbank vorgenommen. Es wird beispielsweise dafür gesorgt, dass das Exemplar nicht mehr in der Bücherbörse ersichtlich ist. Andererseits werden automatisch personalisierte E-Mails an Käufer und Verkäufer versendet. Dieser Code wird bei einem Kauf ausgeführt [34]:

Code-Ausschnitt 26: buch.php

```
if($kaeuer_id == $verkaeuer_id and @$_GET['kaufen']==1){
    echo("<span class='statusmeldung'>Leider kannst du dein eigenes Buch nicht kaufen
:)</span>");
}

//Falls der "Kaufen"-Button betätigt wird, wird der Folgende Code ausgeführt.
elseif(@$_GET['kaufen']==1){
    //Das aktuelle Datum wird in einer Variablen gespeichert,
    $date_time_aktuell = date("Y-m-d H:i:s");
    //Die Bücher-Tabelle wird entsprechend angepasst -> Käufer und Datum werden vermerkt ->
    das Buch wird von nun an nicht mehr in der Bücherbörse ersichtlich sein.
    mysqli_query($con, "UPDATE buecher SET gekauft_von_id = '$kaeuer_id', datum_verkauft =
'$date_time_aktuell' WHERE id='$buch_id'");
    //Aus der "benutzer"-Tabelle werden alle Informationen von Käufer und Verkäufer
    abgefragt.
    $row_verkaeuer = mysqli_fetch_assoc(mysqli_query($con, "SELECT * FROM benutzer WHERE
id='$verkaeuer_id'"));
    $row_kaeuer = mysqli_fetch_assoc(mysqli_query($con, "SELECT * FROM benutzer WHERE
id='$kaeuer_id'"));

    //Die E-Mail-Adressen werden in Variablen gespeichert.
    $email_verkaeuer = $row_verkaeuer['email'];
    $email_kaeuer = $row_kaeuer['email'];

    //Hier werden die E-Mail-Texte definiert, sowohl der Käufer als auch der Verkäufer
    bekommen eine E-Mail mit einer Bestätigung des Verkaufes und den Kontaktdaten des
    jeweils anderen.
    $text_email_verkaeuer = "Hallo " . $row_verkaeuer['vorname'] . "\n\n"
    . "Du hast dein Buch " . $buch_titel . " erfolgreich verkauft. Im Folgenden die
    Kontaktdaten des Käufers:\n\n"
    . $row_kaeuer['vorname'] . " " . $row_kaeuer['nachname'] . "\n"
    . $schulen_titel[array_search($row_kaeuer['schule'], $schulen)] . "\n"
    . $email_kaeuer;

    $text_email_kaeuer = "Hallo " . $row_kaeuer['vorname'] . "\n\n"
    . "Du hast das Buch " . $buch_titel . " erfolgreich gekauft. Im Folgenden die
    Kontaktdaten des Verkäufers:\n\n"
    . $row_verkaeuer['vorname'] . " " . $row_verkaeuer['nachname'] . "\n"
    . $schulen_titel[array_search($row_verkaeuer['schule'], $schulen)] . "\n"
    . $email_verkaeuer;

    //Diese Ausgabe wird dem Benutzer auf der Website nach dem Kauf angezeigt
    echo("<span class='statusmeldung'>Vielen Dank für deinen Kauf! Die Kontaktangaben des
    Verkäufers wurden dir per E-Mail geschickt.</span>");

    //Hier werden die beiden E-Mails versendet.
    mail($email_verkaeuer, "Du hast dein Buch verkauft - kantinet",
    $text_email_verkaeuer);
    mail($email_kaeuer, "Vielen Dank fuer deinen Kauf - kantinet", $text_email_kaeuer);
}

//Hier folgt die standardmässige Ausgabe der Buch-Informationen als Tabelle.
else{
    ...
}
```

Wie die E-Mails, die über den oben aufgeführten Code-Ausschnitt 26 versendet werden, in einem konkreten Beispiel aussehen, wird an den folgenden Abbildungen (Abb. 15 & 16) aufgezeigt. Es handelt sich dabei um Ausschnitte aus einem E-Mail-Programm. Die E-Mail-Adressen der Beispiel-Benutzer wurden zu Testzwecken lokal installiert, daher die Domain «localhost».

Von postmaster@localhost ☆
Betreff **Du hast dein Buch verkauft - kantinet**
An Mich <email1@localhost> ☆

Hallo Marc

Du hast dein Buch Starthilfe Physik erfolgreich verkauft. Im Folgenden die Kontaktdaten des Kaeufers:

Max Mustermann
Kantonsschule Limmattal
email2@localhost

Abb. 15: Der Verkäufer (Marc) hat sein Buch verkauft und erhält eine E-Mail mit den Kontaktdaten des Käufers (Eigenquelle)

Von postmaster@localhost ☆
Betreff **Vielen Dank fuer deinen Kauf - kantinet**
An Mich <email2@localhost> ☆

Hallo Max

Du hast das Buch Starthilfe Physik erfolgreich gekauft. Im Folgenden die Kontaktdaten des Verkaeufers:

Marc Meiners
Kantonsschule Limmattal
email1@localhost

Abb. 16: Der Käufer (Max) hat ein Buch gekauft und erhält eine E-Mail mit den Kontaktdaten des Verkäufers (Eigenquelle)

Die Übergabe der Bücher geschieht individuell, grundsätzlich wird in bar bezahlt. Verkäufer und Käufer tauschen sich via E-Mail aus.

2.6 Dashboard

Ursprünglich wurden grafische Benutzeroberflächen zur Administration von Servern als Dashboard bezeichnet [46]. Heute wird der Begriff jedoch auch oft für Benutzeroberflächen von Websites zur Verwaltung des Benutzerkontos verwendet. So auch im vorliegenden Fall.

Das Dashboard der Schulbuch-Börse hat zwei zentrale Funktionen: Einerseits dient es dem Einsehen und Bearbeiten der Benutzerdaten, andererseits der Verwaltung der Bücher, die einen Bezug zur angemeldeten Person haben (die Person ist Käufer/Verkäufer).

2.6.1 Benutzerdaten

Im oberen Bereich der Benutzerdaten-Ansicht des Dashboards (siehe Abb. 17) werden die Benutzerdaten des eingeloggtten Benutzers mithilfe einer HTML-Tabelle ausgegeben.

kantinet.ch Bücherbörse
Dashboard Logout

Deine Daten

Name	Marc Meiners
Benutzername	marc.meiners
E-Mail-Adresse	email1@localhost
Kantonsschule	Kantonsschule Limmattal
Profil	Physik und Anwendungen der Mathematik
Kontoeröffnung	19.08.2019
Am Verkaufen	10 Bücher
Verkauft	1 Bücher
Verkauft Archiv	1 Bücher
Gekauft	1 Bücher
Gekauft Archiv	1 Bücher

Menu

- ▶ Am Verkaufen
- ▶ Verkauft
- ▶ Verkauft Archiv
- ▶ Gekauft
- ▶ Gekauft Archiv

Ändere deine Daten

Konto löschen

Abb. 17.: Benutzerdaten-Ansicht des Dashboards (Standard-Ansicht) (Eigenquelle: dashboard.php)

2.6.1.1 Benutzerdaten-Änderung

Im unteren Bereich der Benutzerdaten-Ansicht des Dashboards können Benutzername und Passwort geändert werden. Ausserdem kann das Benutzerkonto gelöscht werden.

Für die erwähnten Änderungs-Möglichkeiten sind entsprechend drei HTML-Formulare auf der Seite ersichtlich. Diese bestehen jeweils nur aus einem einzelnen Eingabefeld und einem Submit-Button [29] (siehe Abb. 17).

Wenn eines dieser Formulare abgesendet wird, wird ein POST-Parameter an die Datei zurückgeschickt. Der folgende Code zeigt die Verarbeitung dieser Parameter:

```

Code-Ausschnitt 27:    dashboard.php

//Änderung des Benutzernamens bei entsprechender Anfrage.
if(isset($_POST['neuer_benutzername'])) {
    $neuer_benutzername = mysqli_real_escape_string($con, $_POST['neuer_benutzername']);
    $fehler = false;
    //Kontrolle, ob der Benutzername den Zeichenrichtlinien entspricht.
    if (strlen($neuer_benutzername)>25 or strlen($neuer_benutzername)<5 ) {
        echo("<span class='statusmeldung'>Der Benutzername muss zwischen 5 und 25
        Zeichen lang sein</span> </br>");
        $fehler=true;
    }
}

```



```

//Kontrolle, ob Benutzername bereits in der Datenbank existiert.
if (mysqli_num_rows(mysqli_query($con, "SELECT benutzername FROM benutzer WHERE
benutzername='$neuer_benutzername'"))==1){
echo("<span class='statusmeldung'>Der Benutzername existiert bereits. Bitte
wähle einen anderen.</span></br>");
$fehler=true;
}
//Datenbank-Aktualisierung.
if ($fehler == false){
mysqli_query($con, "UPDATE benutzer SET benutzername = '$neuer_benutzername',
datum_aktualisiert = '$date_time_aktuell' WHERE id = '$benutzer_id'");
echo("<span class='statusmeldung'>Du hast deinen Benutzernamen erfolgreich
geändert!</span>");
}
}

//Änderung des Passworts bei entsprechender Anfrage.
if(isset($_POST['neues_passwort'])){
$neues_passwort = mysqli_real_escape_string($con, $_POST['neues_passwort']);
$fehler = false;
//Kontrolle, ob das neue Passwort den Zeichenrichtlinien entspricht.
if (strlen($neues_passwort)>50 or strlen($neues_passwort)<8 ){
echo("<span class='statusmeldung'>Das Passwort muss zwischen 8 und 50 Zeichen
lang sein</span></br>");
$fehler=true;
}
if($fehler==false){
//Umwandlung des neuen Passwortes in einen Hash.
$neues_passwort_hash = sha1($neues_passwort);
//Datenbank-Aktualisierung.
mysqli_query($con, "UPDATE benutzer SET password = '$neues_passwort_hash',
datum_aktualisiert = '$date_time_aktuell' WHERE id = '$benutzer_id'");
echo("<span class='statusmeldung'>Du hast dein Passwort erfolgreich
geändert!</span>");
}
}

//Kontolöschung bei entsprechender Anfrage.
if(isset($_POST['konto_loeschen'])){
//Damit die Kontolöschung nicht aus Versehen erfolgt, muss der Benutzer im
entsprechenden Dashboard-Formular die Löschung durch Eingabe einer Kennung ("1234")
bestätigen.
if($_POST['konto_loeschen'] == "1234"){
//Das Konto wird als gelöscht definiert, der Datenbankeintrag bleibt vorerst
erhalten.
mysqli_query($con, "UPDATE benutzer SET datum_geloescht = '$date_time_aktuell'
WHERE id = '$benutzer_id'");
//Alle Bücher, die vom Benutzer jemals zum Verkauf eingestellt wurden, werden
als gelöscht markiert und nicht mehr in der Bücherbörse angezeigt.
mysqli_query($con, "UPDATE buecher SET benutzer_geloescht = '1' WHERE
benutzer_id = '$benutzer_id'");
//Löschen der "Benutzer-Session" und Weiterleitung zur Startseite.
session_destroy();
header("Location: index.php");
}
else{
echo("Wenn du das Benutzerkonto löschen möchtest, dann gib bitte '1234' in das
Formular ein.");
}
}
}

```

2.6.2 Bücherverwaltung

2.6.2.1 Sidebar

Die Sidebar des Dashboards mit dem Titel «Menu» (siehe Abb. 17) funktioniert analog zu derjenigen der Bücher-Verwaltung (vgl. Kap. 2.5.2.1) und dient als eine Art Navigationsleiste für die fünf Kategorien der Bücher-Verwaltung.

2.6.2.2 Grundprinzip der fünf Kategorien

Die Bücher-Verwaltung des Dashboards ist in fünf Kategorien unterteilt. Diese werden im Folgenden aufgezeigt, wobei jeweils der obere Teil der Seiten als Abbildung (Abb. 16-20) ersichtlich ist, sodass das oberste Buch der Kategorie ersichtlich ist. Darunter können

weitere Bücher aufgelistet sein. Die Einträge werden von horizontalen Strichen [30] voneinander getrennt (siehe Abb. 18).

- **Am Verkaufen:** Hier werden die aktuell vom angemeldeten Benutzer registrierten zum Verkauf angebotenen Bücher angezeigt. Es besteht die Möglichkeit, ein bestimmtes Buch über einen Button aus der Bücherbörse zu entfernen.



Abb. 18: Dashboard-Menü "Am Verkaufen" (Eigenquelle: dashboard.php)

- **Verkauft:** Wurde bei einem der registrierten Bücher des angemeldeten Benutzers seitens eines anderen Benutzers der «Kaufen»-Button in der Buchdetails-Ansicht (vgl. Kap. 2.5.3) betätigt, wird das Buch in die Kategorie «Verkauft» verschoben. Nun hat der angemeldete Benutzer die Möglichkeiten, den Verkaufsprozess als abgeschlossen zu kennzeichnen. Wenn der Verkauf nicht erfolgreich war, beispielsweise, wenn der Käufer einen Rückzug gemacht hat, kann das Exemplar über einen entsprechenden Button wieder online gestellt werden. Aus diesem Grund gibt es diese Zwischenstufe, wobei das Exemplar erst bei einem endgültigen Verkauf ins Archiv kommt.



Abb. 19: Dashboard-Menü "Verkauft" (Eigenquelle: dashboard.php)

- **Verkauft Archiv:** Hier werden die verkauften Bücher angezeigt, bei denen der Verkaufsprozess abgeschlossen ist.



Abb. 20: Dashboard-Menü "Verkauft Archiv" (Eigenquelle: dashboard.php)

- **Gekauft:** Hier werden die Bücher angezeigt, die der angemeldete Benutzer selbst gekauft hat, bei denen der Verkaufsprozess aber noch nicht abgeschlossen ist. Bei den entsprechenden Verkäufern werden dieselben Exemplare im «Verkauft»-Menü angezeigt.



Abb. 21: Dashboard-Menü "Gekauft" (Eigenquelle: dashboard.php)

- **Gekauft Archiv:** Hier werden die gekauften Bücher angezeigt, bei denen der Verkaufsprozess abgeschlossen ist. Nur der Verkäufer kann Bücher ins Archiv verschieben.



Abb. 22: Dashboard-Menü "Gekauft Archiv" (Eigenquelle: dashboard.php)

2.6.2.3 Hintergrund-Technik

2.6.2.3.1 Datenbankabfragen

Mit den folgenden Datenbankabfragen können die Informationen der Bücher in den jeweiligen Kategorien (vgl. Kap. 2.6.2.1) ausgegeben werden:

```
Code-Ausschnitt 28:    dashboard.php

$res_am_verkaufen = mysqli_query($con, "SELECT * FROM buecher WHERE benutzer_id =
'$benutzer_id' AND gekauft_von_id = '0' AND datum_verkauf_abgeschlossen = '0000-00-00
00:00:00' ORDER BY id DESC");
$res_verkauft = mysqli_query($con, "SELECT * FROM buecher WHERE benutzer_id = '$benutzer_id'
AND gekauft_von_id != '0' AND datum_verkauf_abgeschlossen = '0000-00-00 00:00:00' ORDER BY id
DESC");
$res_verkauft_archiv = mysqli_query($con, "SELECT * FROM buecher WHERE benutzer_id =
'$benutzer_id' AND gekauft_von_id != '0' AND datum_verkauf_abgeschlossen != '0000-00-00
00:00:00' ORDER BY id DESC");
$res_gekauft = mysqli_query($con, "SELECT * FROM buecher WHERE gekauft_von_id = '$benutzer_id'
AND datum_verkauf_abgeschlossen = '0000-00-00 00:00:00' ORDER BY id DESC");
$res_gekauft_archiv = $res = mysqli_query($con, "SELECT * FROM buecher WHERE gekauft_von_id =
'$benutzer_id' AND datum_verkauf_abgeschlossen != '0000-00-00 00:00:00' ORDER BY id DESC");
```

Anschliessend wird analog zur Bücher-Übersicht (vgl. Kap. 2.5.2.2) eine While-Schleife verwendet, um die einzelnen Datensätze der Abfragen in den jeweiligen Menüs beziehungsweise Kategorien der Bücherverwaltung untereinander aufzulisten.

2.6.2.3.2 Aktionen in der Bücherverwaltung

Bei den in Kapitel 2.6.2.2 aufgezeigten Menüs sind in den Kategorien «Am Verkaufen» und «Verkauft» (vgl. Abb. 16 & 17) unter den dort aufgelisteten Büchern Buttons ersichtlich. Diese lösen die entsprechende Aktion, mit der sie betitelt sind, aus, indem beim Klicken ein Formular an dieselbe Datei zurückgeschickt wird. Hier wird aufgezeigt, wie die übermittelten POST-Parameter verarbeitet werden:

Code-Ausschnitt 29: dashboard.php

```
if(isset($_POST['buch_loeschen'])){
    $buch_loeschen_id = $_POST['buch_loeschen'];
    mysqli_query($con, "DELETE FROM buecher WHERE id = '$buch_loeschen_id'");
    echo("<span class='statusmeldung'>Du hast dein Buch erfolgreich gelöscht. Es wird nun
    nicht mehr auf der Website gelistet.</span>");
}

//Ein noch nicht abgeschlossener Verkaufsprozess kann vom Verkäufer abgebrochen werden.
//Das Buch wird in diesem Fall reaktiviert und ist wieder in der Bücherbörse ersichtlich.
if(isset($_POST['buch_reaktivieren_id'])){
    $buch_reaktivieren_id = $_POST['buch_reaktivieren_id'];
    mysqli_query($con, "UPDATE buecher SET datum_verkauft = '0000-00-00 00:00:00',
    gekauft_von_id = '0' WHERE id = '$buch_reaktivieren_id'");
    echo("<span class='statusmeldung'></span>");
}

//Der abgeschlossene Verkaufsprozess kann nur seitens des Verkäufers bestätigt werden.
//Das entsprechende Buch erscheint nun bei Käufer und Verkäufer im Archiv.
if(isset($_POST['buch_archiv_id'])){
    $buch_archiv_id = $_POST['buch_archiv_id'];
    mysqli_query($con, "UPDATE buecher SET datum_verkauf_abgeschlossen =
    '$date_time_aktuell' WHERE id = '$buch_archiv_id'");
    echo("<span class='statusmeldung'></span>");
}
```

2.7 Schutz vor SQL-Injections

Auf der Schulbuch-Börse sind zahlreiche Eingabefelder und Link-Parameter vorhanden, die in einem späteren Verarbeitungsschritt des Programms eine Datenbankabfrage spezifizieren. Diese Tatsache können allfällige Hacker ausnutzen, indem sie beispielsweise über ein Eingabefeld SQL-Code einschleusen, um beliebige Datenbankeinträge, wie Benutzerdaten, auslesen oder verändern zu können. Diese Sicherheitslücke besteht bei mangelnder Überprüfung der vom Benutzer eingegebenen Daten [47].

Um dem Problem entgegenzusteuern hat PHP eigens eine Funktion bereitgestellt, die Zeichenketten auf SQL-Befehle überprüft und entsprechende spezielle Zeichen oder Zeichenfolgen entfernt [2].

Das folgende Beispiel zeigt einen Teil der Datei, die für das Login zuständig ist. Hier wird die Funktion zweimal angewendet, um den vom Benutzer eingegebenen Benutzernamen und das Passwort zu überprüfen:

Code-Ausschnitt 30: login.php

```
//Definieren von später genutzten Variablen.
$benutzer = mysqli_real_escape_string($con, $_POST['benutzer']);
$password = mysqli_real_escape_string($con, $_POST['password']);
```

3 Schlusswort

3.1 Zusammenfassung

Das Ziel meiner Maturitätsarbeit war es, eine Schulbuch-Börse für Zürcher Kantonsschüler zu programmieren. Ich habe mich dabei an die in der Einleitung geschilderten Ziele gehalten und diese erfüllt.

Ich habe ein Instrument entwickelt, das in voller Interaktivität den Verkauf und Kauf von Schulbüchern über eine zentrale Plattform ermöglicht. Ergänzt habe ich das ursprüngliche Vorhaben beispielsweise durch eine API, die automatisch Buch-Informationen abrufen und den Registrierungsprozess vereinfacht. Des Weiteren habe ich die Bücher-Übersicht durch einen Filter und eine Blätter-Möglichkeit ergänzt, sodass das Tool in seiner Gesamtheit übersichtlicher wird und der Käufer schneller das gewünschte Exemplar findet.

3.2 Schwierigkeiten

Ganz allgemein auf die Maturitätsarbeit bezogen, denke ich, dass es vielen Schülern schwerfällt, ein Thema auszusuchen, das in der vorgegebenen Zeit realisierbar ist und den Rahmen nicht sprengt. Ebendieses Problem hatte auch ich: Anfangs hatte ich noch Ideen, wie ich mein Online-Tool ergänzen könnte. Beispielsweise durch einen Vokabel-Trainer oder ein Forum. Glücklicherweise habe ich sehr früh gemerkt, dass diese Vorhaben aus zeitlichen Gründen unrealistisch sind. Stattdessen habe ich mich auf die Schulbuch-Börse konzentriert und schnell gemerkt, dass deren Umsetzung den Zeitrahmen gut ausfüllen würde. Ich habe nach dem Motto entschieden, dass ich mir lieber weniger vornehme, das geplante schliesslich aber auch umsetze.

Eine weitere Hürde war die schriftliche Arbeit. Es war in vielen Fällen nicht einfach, einen Arbeitsschritt oder ein Element der Schulbuch-Börse mit Worten zu beschreiben. Auch war es oft eine Herausforderung, abzuwägen, welche Teile meiner praktischen Arbeit ich wie ausführlich im schriftlichen Teil schildern sollte. Hätte ich jeden Arbeitsschritt und Code-Ausschnitt genauestens dokumentiert, wäre die Seitenzahl der vorliegenden Arbeit wohl im dreistelligen Bereich zu liegen gekommen. Deshalb galt es, ein gewisses Mittelmass zu finden, wobei mir mein Betreuer eine gute Hilfe war, indem er mir diesbezüglich nach dem Lesen der ersten Seiten eine kurze Rückmeldung gab.

3.3 Mögliche Weiterentwicklungen des Projekts

Ein nächster Schritt könnte sein, die Website online zu stellen. Dabei stellt sich allerdings die Frage, ob ich die Seite zuvor, gerade bezogen auf ihr Design, welches nicht im Zentrum dieser Arbeit stand, auffrischen sollte. Die meisten Kantonsschüler, da bin ich mir sicher, würden die Seite mit ihrem Smartphone aufrufen. Meine Website ist nicht «responsive», also nicht auf verschiedene Gerätegrössen zugeschnitten, was die Bedienung des Tools mit einem mobilen Gerät mühsam macht.

Ein weiterer Punkt ist, dass die Regelung des Copyrights der Cover-Bilder aus der Google Books API unklar ist: Vor einer Veröffentlichung müssten die rechtlichen Grundlagen abgeklärt und entsprechende Anmerkungen in der Schulbuch-Börse hinterlegt werden.

In der aktuellen Version der Website läuft der Reaktivierungs-Prozess nur bei Login-Vorgängen ab. Eine Alternative wäre ein serverseitiges Script, welches beispielsweise einmal am Tag alle Benutzerkonten auf deren Reaktivierungsstatus überprüft und entsprechend E-Mails versendet und Konten deaktiviert. Für die Realisierung dieser Idee wäre allerdings das Erlernen einer weiteren höheren Programmiersprache von Nöten, um

das erwähnte Script umzusetzen. Dies wäre im Rahmen dieser Maturitätsarbeit zu umfangreich gewesen.

3.4 Erkenntnisse aus der Arbeit

Wenn ich darüber nachdenke, was mir der Arbeitsprozess der Maturitätsarbeit für meine Zukunft gebracht hat, fällt mir als erstes der Begriff Zeitmanagement ein. Diese simple Aussage ist umfassender, als man vielleicht denkt. Ich habe das erste Mal in meinem Leben ein Projekt mit einem Umfang von 90 Arbeitsstunden, für das ich ein Jahr Zeit hatte, völlig selbstständig realisiert und musste mir meine Zeit selbst einteilen. Gerade die Tatsache, dass ich Aufgaben und Pflichten in der Regel im letzten Moment zu tun pflege, stellte mich vor eine Herausforderung.

Des Weiteren habe ich durch Learning by Doing meine Programmierkenntnisse verbessert und praktische Erfahrungen im Entwickeln und Planen eines grösseren Projekts gesammelt. Gerade der richtige Umgang mit auftretenden Schwierigkeiten und das strukturierte Herangehen an Probleme habe ich auf praktische Weise lernen können.

4 Literaturverzeichnis

Anmerkungen:

1. Alle der im Folgenden aufgelisteten Text-Quellen, ausser die erste, wurden einmalig an der jeweils entsprechenden Stelle im normal formatierten Fliesstext dieser Arbeit vermerkt.
2. Die erste Text-Quelle ([1]) ist im Gegensatz zu allen anderen keine Internet-Quelle, sondern eine Buch-Quelle. Das Buch als Ganzes wurde zum Aufbau von Grundlagenwissen zugezogen, weshalb keine Seitenzahl angegeben ist und die Quelle an keiner Stelle der Arbeit vermerkt wurde.

- [1] Thomas Theis, *Einstieg in PHP 7 und MySQL. Ideal für Programmieranfänger*, 13. Auflage, Bonn, Rheinwerk Verlag, 2018
- [2] PHP `mysqli_real_escape_string()` Function, *W3schools.com*. [Online] abrufbar: https://www.w3schools.com/php/func_mysqli_real_escape_string.asp (Abrufdatum: 18.08.2019)
- [3] `Strlen`, *Php.net*. [Online] abrufbar: <https://www.php.net/manual/de/function.strlen.php> (Abrufdatum: 29.04.2019)
- [4] Prüfen, ob ein bestimmter Datensatz vorhanden ist, *PHP Performance*. [Online] abrufbar: <https://phpperformance.de/web-dev/php/pruefen-ob-ein-bestimmter-datensatz-vorhanden-ist> (Abrufdatum: 30.04.2019)
- [5] Splitting strings in PHP and get last part, *Stack overflow*. [Online] abrufbar: <https://stackoverflow.com/questions/17030779/splitting-strings-in-php-and-get-last-part> (Abrufdatum: 29.04.2019)
- [6] PHP `filter_var()` Function, *W3schools.com*. [Online] abrufbar: https://www.w3schools.com/php/func_filter_var.asp (Abrufdatum: 29.04.2019)
- [7] How to use Mercury Mail in XAMPP, *New Coder's Life*. [Online] abrufbar: <https://newcoderslife.wordpress.com/2012/06/27/how-to-use-xampps-mail-server-mercury-mail/> (Abrufdatum: 30.04.2019)
- [8] Verschlüsseln und Entschlüsseln in PHP, *A Coding Project*. [Online] abrufbar: <https://www.a-coding-project.de/ratgeber/php/verschluesseln-und-entschluesseln> (Abrufdatum: 29.04.2019)
- [9] Speichern php datetime in mysql Datenbank, *Stack overrun*. [Online] abrufbar: <https://stackoverrun.com/de/q/3688298> (Abrufdatum: 29.04.2019)
- [10] Get a PHP DateTime difference in days, considering midnight as a day change, *Stack overflow*. [Online] abrufbar: <https://stackoverflow.com/questions/18102603/get-a-php-datetime-difference-in-days-considering-midnight-as-a-day-change> (Abrufdatum: 03.05.2019)
- [11] Sessions, *PHP-Einfach.de*. [Online] abrufbar: <https://www.php-einfach.de/php-tutorial/php-sessions/> (Abrufdatum: 30.04.2019)
- [12] PHP String to Float, *Stack overflow*. [Online] abrufbar: <https://stackoverflow.com/questions/481466/php-string-to-float> (Abrufdatum: 02.05.2019)
- [13] Mit PHP einen Betrag auf 5 Rappen runden, *PAS solutions GmbH*. [Online] abrufbar: <https://www.pas-solutions.ch/webdesign/mit-php-einen-betrag-auf-5-rappen-runden/> (Abrufdatum: 02.05.2019)
- [14] `number_format`, *Php.net*. [Online] abrufbar: <https://www.php.net/manual/de/function.number-format.php> (Abrufdatum: 02.05.2019)
- [15] Using the API, *Google Books APIs*. [Online] abrufbar: <https://developers.google.com/books/docs/v1/using> (Abrufdatum: 01.05.2019)
- [16] PHP `str_replace()` Function, *W3schools.com*. [Online] abrufbar: https://www.w3schools.com/php/func_string_str_replace.asp (Abrufdatum: 01.05.2019)
- [17] Die Ersten 5 Zeichen eines Strings, *Selfphp.de*. [Online] abrufbar: <http://www.selfphp.de/forum/showthread.php?t=8364> (Abrufdatum: 02.05.2019)
- [18] Ersatzbild in HTML anzeigen, wenn Bild nicht gefunden wurde, *Interaktive Fundgrube*. [Online] abrufbar: <http://www.interaktive-fundgrube.de/2008/05/26/ersatzbild-in-html-anzeigen-wenn-bild-nicht-gefunden-wurde/> (Abrufdatum: 16.07.2019)
- [19] Prüfen, ob im Array ein Wert vorhanden ist, *RIP Tutorial*. [Online] abrufbar: <https://riptutorial.com/de/php/example/3181/prufen--ob-im-array-ein-wert-vorhanden-ist> (Abrufdatum: 16.07.2019)
- [20] `Intval`, *Php.net*. [Online] abrufbar: <https://www.php.net/manual/de/function.intval.php> (Abrufdatum: 15.07.2019)
- [21] LIMIT bei MySQL Abfragen, *Backslash.ch*. [Online] abrufbar: <https://www.backslash.ch/blog/2007/6/6/LIMIT-bei-MYSQL-Abfragen> (Abrufdatum: 15.07.2019)
- [22] Anzeigen der abgefragten Daten, *PHP für dich*. [Online] abrufbar: <https://www.schattenbaum.net/php/anzeigen.php> (Abrufdatum: 15.07.2019)
- [23] Max-height, *MDN web docs*. [Online] abrufbar: <https://developer.mozilla.org/de/docs/Web/CSS/max-height> (Abrufdatum: 15.07.2019)
- [24] Bilder mit CSS horizontal zentrieren, *Lewumpy*. [Online] abrufbar: <https://blog.lewumpy.de/2009/09/bilder-mit-css-horizontal-zentrieren/> (Abrufdatum: 15.07.2019)
- [25] Sortieren mit MySQL ORDER BY, *Use-media*. [Online] abrufbar: <https://www.use-media.com/sortieren-mit-mysql-order-by.html> (Abrufdatum: 15.07.2019)
- [26] CSS word-wrap – lange Wörter automatisch umbrechen, *CSS, HTML und Javascript mit stil*. [Online] abrufbar: <https://www.mediaevent.de/css/word-wrap.html> (Abrufdatum: 18.07.2019)
- [27] HTML Sonderzeichen – Favoriten, *Vioma*. [Online] abrufbar: <https://www.vioma.de/de/wiki/tools/html-sonderzeichen/#HTML%20Sonderzeichen%20-%20Favoriten> (Abrufdatum: 16.07.2019)
- [28] Die aktuelle URL ermitteln, *Rither*. [Online] abrufbar: <http://www.rither.de/a/informatik/php-beispiele/sonstiges/die->

aktuelle-url-ermitteln/ (Abrufdatum: 16.07.2019)

[29] Text von button linksbündig, *Tutorials.de*. [Online] abrufbar: <https://www.tutorials.de/threads/text-von-submit-button-links-buendig.187921/> (Abrufdatum: 23.07.2019)

[30] How TO - Style HR, *W3schools.com*. [Online] abrufbar: https://www.w3schools.com/howto/howto_css_style_hr.asp (Abrufdatum: 19.07.2019)

[31] Liste der Kantonsschulen des Kantons Zürich, *Wikipedia*. [Online] abrufbar: https://de.wikipedia.org/wiki/Liste_der_Kantonsschulen_des_Kantons_Z%C3%Bcrich (Abrufdatum: 25.07.2019)

[32] Falsche Ausgabe von Umlauten unter MySQL, *Administrator*. [Online] abrufbar: <https://administrator.de/forum/falsche-ausgabe-umlauten-mysql-161431.html> (Abrufdatum: 02.05.2019)

[33] Array_search, *Php.net*. [Online] abrufbar: <https://www.php.net/manual/de/function.array-search.php> (Abrufdatum: 18.07.2019)

[34] Zeilenumbrüche bei Email, *Tutorials.de*. [Online] abrufbar: <https://www.tutorials.de/threads/zeilenumbrueche-bei-email.372555/> (Abrufdatum: 18.07.2019)

[35] CSS Tables, *W3schools.com*. [Online] abrufbar: https://www.w3schools.com/css/css_table.asp (Abrufdatum: 18.07.2019)

[36] Button als Link?, *HTML-seminar.de*. [Online] abrufbar: <http://www.html-seminar.de/forum/thread/7561-button-als-link/?postID=51841> (Abrufdatum: 18.07.2019)

[37] PHP: Aktuelle Jahreszahl ausgeben, *Computer-Masters.de*. [Online] abrufbar: <http://www.computer-masters.de/php-aktuelle-jahreszahl-ausgeben.php> (Abrufdatum: 17.07.2019)

[38] Hypertext Markup Language, *Wikipedia*. [Online] abrufbar: https://de.wikipedia.org/wiki/Hypertext_Markup_Language (Abrufdatum: 01.09.2019)

[39] Cascading Style Sheets, *Wikipedia*. [Online] abrufbar: https://de.wikipedia.org/wiki/Cascading_Style_Sheets (Abrufdatum: 01.09.2019)

[40] PHP, *Wikipedia*. [Online] abrufbar: <https://de.wikipedia.org/wiki/PHP> (Abrufdatum: 01.09.2019)

[41] MySQL, *Wikipedia*. [Online] abrufbar: <https://de.wikipedia.org/wiki/MySQL> (Abrufdatum: 01.09.2019)

[42] SQL, *Wikipedia*. [Online] abrufbar: <https://de.wikipedia.org/wiki/SQL> (Abrufdatum: 01.09.2019)

[43] XAMPP, *Wikipedia*. [Online] abrufbar: <https://de.wikipedia.org/wiki/XAMPP> (Abrufdatum: 05.09.2019)

[44] Internationale Standardbuchnummer, *Wikipedia*. [Online] abrufbar: https://de.wikipedia.org/wiki/Internationale_Standardbuchnummer (Abrufdatum: 14.09.2019)

[45] Adobe Dreamweaver, *Wikipedia*. [Online] abrufbar: https://de.wikipedia.org/wiki/Adobe_Dreamweaver (Abrufdatum: 08.10.2019)

[46] Dashboard (IT-Administration), *Wikipedia*. [Online] abrufbar: [https://de.wikipedia.org/wiki/Dashboard_\(IT-Administration\)](https://de.wikipedia.org/wiki/Dashboard_(IT-Administration)) (Abrufdatum: 09.10.2019)

[47] SQL-Injection, *Wikipedia*. [Online] abrufbar: <https://de.wikipedia.org/wiki/SQL-Injection> (Abrufdatum: 10.10.2019)

5 Abbildungsverzeichnis

[48] Book-cover empty, *Pixabay*. [Online] abrufbar: <https://pixabay.com/illustrations/book-isolated-book-cover-empty-3057901/> (Abrufdatum: 01.05.2019)

[49] Darstellung der Funktionsweise von PHP, *Wikipedia*. [Online] abrufbar: https://de.wikipedia.org/wiki/PHP#/media/Datei:PHP_funktionsweise.svg (Abrufdatum: 05.09.2019)

[50] *Google Books*. [Online] abrufbar: <http://books.google.com/books> (verschiedene Abrufdaten im Jahr 2019)

6 Einhaltung rechtlicher Vorgaben

Ich habe die Arbeit selbstständig und unter Aufsicht meines Betreuers verfasst und keine anderen als die angegebenen Hilfsmittel verwendet.

Ich nehme zur Kenntnis, dass meine Arbeit zur Überprüfung der korrekten und vollständigen Angabe der Quellen mit Hilfe einer Software (eines Plagiaterkennungstools) geprüft wird. Zu meinem eigenen Schutz wird die Software auch dazu verwendet, später eingereichte Arbeiten mit meiner Arbeit elektronisch zu vergleichen und damit Abschriften und eine Verletzung meines Urheberrechts zu verhindern.

Falls Verdacht besteht, dass mein Urheberrecht verletzt wurde, erkläre ich mich damit einverstanden, dass die Schulleitung meine Arbeit zu Prüfzwecken herausgibt.

Datum, Unterschrift
