

Hochschule Hannover

Fakultät III –Information & Design

Abteilung Information und Kommunikation

Konzeption und Anwendung eines WordPress-Plugins für den Vergleich von Veranstaltungen

Bachelorarbeit

im Studiengang Informationsmanagement

vorgelegt von

Lukas Fischer

Erstgutachterin: Prof. Dr.-Ing. Monika Steinberg

Zweitgutachter: Prof. Dr. Thomas J. Schult

Hannover, den 06.07.2020

Abstract

Plugins erweitern die Funktionalität von WordPress und helfen Webseitenbetreibern beim Hinzufügen neuer Elemente oder Funktionen. Dabei muss der Betreiber selbst keinen komplexen Programmcode schreiben. Auch können diese Erweiterungen als Erleichterung bei der Pflege von Informationen und Inhalten dienen. In der vorliegenden Bachelorarbeit wird die Konzipierung, Umsetzung und Anwendung eines solchen Plugins für den Vergleich von Veranstaltungen beschrieben. Dabei findet es Anwendung auf einer Internetseite zum Thema Festivals. Die Informationen bezüglich der Festivals werden per Import in einer Tabelle gespeichert und für den Vergleich dargestellt. Eintragungen können auch händisch getätigt werden, woraus sich schließlich ergibt, dass das Plugin mit seiner Importfunktion eine zeitsparende und sinnvolle Erweiterung ist.

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Quellcodeverzeichnis	V
Einleitung	1
1 Anwendungsbereich	3
1.1 WordPress	3
1.2 Voraussetzungen	4
1.3 Konzertkalender.net	5
1.3.1 Aufbau der Seite	5
1.3.2 Aufbau einer Veranstaltung	6
1.3.3 Funktionen	7
2 Problemstellung	11
3 Konzept	12
3.1 Aufbau	12
3.1.1 Frontend	12
3.1.2 Menü	12
3.1.3 Installierte Plugins	13
3.2 Funktionen	13
3.2.1 Import	13
3.2.2 Einfügen	13
3.2.3 Löschen	14
3.2.4 Shortcode	14
4 Komponenten	15
4.1 Hypertext Markup Language	15
4.2 Cascading Style Sheets	15
4.3 PHP: Hypertext Preprocessor	16
4.4 Structured Query Language	16
5 Implementierung	18
5.1 Aufbau	18
5.2 Dateien	18
5.2.1 index.php	19
5.2.2 uninstall.php	25
5.2.3 connection.php	25
5.2.4 importdaten.php	26
5.2.5 dbtabelle.php	30
5.2.6 ausgeben.php	31

5.2.7	tabelle.php	32
5.2.8	tabelle.css	35
5.2.9	datumformat.php.....	37
5.2.10	tage.php.....	37
5.2.11	startseite.php.....	38
6	Auswertung	39
6.1	Entwurf und Realisierung	39
6.2	Testlauf	39
6.3	Veröffentlichung	42
	Fazit und Ausblick	43
	Literaturverzeichnis	45
	Eidesstattliche Erklärung.....	48
	Anhang 1: index.php.....	49
	Anhang 2: uninstall.php	52
	Anhang 3: ausgeben.php	52
	Anhang 4: connection.php.....	53
	Anhang 5: datumformat.php.....	53
	Anhang 6: dbtabelle.php	53
	Anhang 7: importdaten.php.....	54
	Anhang 8: startseite.php.....	57
	Anhang 9: tabelle.php	58
	Anhang 10: tage.php.....	66
	Anhang 11: tabelle.css	66

Abkürzungsverzeichnis

CMS	Content-Management-System
CPT	Custom Post Type
CSS	Cascading Style Sheets
DCL	Data Control Language
DDL	Data Definition Language
DML	Data Manipulation Language
DQL	Data Query Language
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
PHP	PHP: Hypertext Preprocessor
SEO	Suchmaschinenoptimierung
SQL	Structured Query Language
URL	Uniform Resource Locator
WWW	World Wide Web

Abbildungsverzeichnis

Abbildung 1: konzertkalender.net	6
Abbildung 2: Veranstaltungsseite	7
Abbildung 3: Beispiel Schema.org (Google)	9
Abbildung 4: Entwurf der Tabelle	12
Abbildung 5: Menü Entwurf	13
Abbildung 6: Funktionsweise PHP	16
Abbildung 7: Sprachumfang von Befehlsgruppen in SQL.....	17
Abbildung 8: Aufbau Plugin	18
Abbildung 9: Aufbau Importfunktion.....	26
Abbildung 10: Datenbanktabelle erstellen.....	31
Abbildung 11: Funktion Tabelle ausgeben	32
Abbildung 12: Vergleichstabelle im Frontend	34
Abbildung 13: Vergleichstabelle (Handy).....	36
Abbildung 14: Funktion Datum umwandeln	37
Abbildung 15: Tage errechnen	38
Abbildung 16: Anleitung des Plugins im Backend.....	38
Abbildung 17: Eintrag in Liste der installierten Plugins	39
Abbildung 18: Eigenen Eintrag erstellen	40
Abbildung 19: Liste der importierten Festivals.....	40
Abbildung 20: Vergleichstabelle im Frontend	41
Abbildung 21: Liste der importieren Festivals.....	41
Abbildung 22: Vergleichstabelle im Frontend	42

Quellcodeverzeichnis

Quellcode 1: Plugin-Kopfzeilenkommentar	19
Quellcode 2: Menü einfügen	21
Quellcode 3: Hinzufügen eines CPT	22
Quellcode 4: Meta-Box hinzufügen	23
Quellcode 5: Shortcode hinzufügen.....	24
Quellcode 6: Plugin CSS hinzufügen.....	24
Quellcode 7: uninstall.php	25
Quellcode 8: SQL-Befehl für den Import.....	27
Quellcode 9: SQL-Befehl für Hinzufügen von Informationen beim Import.....	28
Quellcode 10: SQL-Befehl für den Import der eigenen Festivals.....	29
Quellcode 11: SQL-Befehl zum Löschen von Einträgen	29
Quellcode 12: SQL-Befehl zum Zählen der Datensätze in der Tabelle.....	29
Quellcode 13: Aktualisierungsfunktion der Einträge	30
Quellcode 14: Datenbanktabelle wird in ein Array geschrieben.....	33
Quellcode 15: Auswahloptionen für die Tabelle	33
Quellcode 16: Ausgabe der Tabelle	34
Quellcode 17: tabelle.css	35

Einleitung

Das Ziel eines Internetseitenbetreibers ist es die eigene Seite bestmöglich von anderen Seiten mit ähnlichem Inhalt abzuheben und so eine möglichst hohe Anzahl von Nutzern für sich zu gewinnen. Bei der als Vorarbeit entstandenen Internetseite geht es um eine mit dem Content-Management-System (CMS) WordPress umgesetzte Webseite, die Musikfestivals mit den dazugehörigen Informationen als Veranstaltung darstellt.¹ Dargestellt werden u. a. der Zeitraum der Veranstaltung, die Preise, die Adresse und Weiterleitungen zum Ticketkauf. Um die Seite stetig auszubauen, zu verbessern und von anderen ähnlichen Seiten abzusetzen, sind fortwährend neuen Funktionen notwendig.

Ein Vergleich kleiner Festivals, z. B. als Entscheidungshilfe für den nächsten Besuch, wird mangels verfügbarer Vergleichsmöglichkeiten zum mühsamen Zusammentragen vereinzelter Informationen von verschiedenen Internetseiten. Eine Erweiterung, die diese Informationen auf einer einzelnen Seite sammelt und darstellt, würde diesen Vorgang erheblich beschleunigen und vereinfachen. Zudem sorgt es für ein Alleinstellungsmerkmal der Webseite. Die Realisierung der Erweiterung ist in Form eines WordPress-Plugins umsetzbar. Bei einem Plugin handelt es sich um, ein von WordPress abhängiges Programm, welches das CMS mit bestimmten Funktionen erweitern kann.² Das fertige Plugin soll eine Vergleichstabelle generieren und bereits in WordPress erstellte Veranstaltungen in eine Tabelle per Klick importieren, ohne dass diese Informationen vom Nutzer erneut eingetragen werden müssen.

WordPress hat den höchsten Marktanteil der Content Management Systeme im Januar 2020³, dementsprechend ist auch die Anzahl der Erweiterungen sehr umfangreich. Ob vom Erlernen, dem fortgeschrittenen Programmieren von und mit PHP, HTML und CSS, bis hin zu detaillierten Leitfäden und Tutorials sowie Beschreibungen zu WordPress, ist alles vorhanden. Darüber hinaus besteht eine große Online-Community. In Bezug auf Tabellen-Plugins für WordPress, die das Vergleichen von Werten möglich machen, bedeutet das, dass dementsprechend bereits viele Varianten vorhanden sind. Allerdings müssen die Informationen alle händisch eingetragen werden. Angewendet auf das zu Anfang beschriebene Projekt hätte das eine redundante Datenhaltung und einen wesentlich größeren Aufwand zufolge, da

¹ www.konzertkalender.net

² Vgl. Kommer 2019, S.87

³ Vgl. Statista.com 2020

die Festivals bereits als Veranstaltung existieren und somit noch zusätzlich per Hand eingetragen werden müssten.

Im Rahmen dieser Arbeit wird ein WordPress-Plugin realisiert, das bereits vorhandene Veranstaltungen möglichst unkompliziert und schnell in eine generierte Tabelle einfügt, ohne dass der Benutzer diese doppelt eintragen müsste. Manuelle Eintragungen sollen jedoch ebenfalls durchführbar sein. Die eingefügten Informationen können dann übersichtlich miteinander verglichen werden. Bevor das Ganze umgesetzt werden kann, muss die Funktion und das Zusammenspiel des Plugins mit der Seite geklärt werden. Als Vorlage dient hierzu ein zuvor erarbeitetes Konzept, aus dem Schritt für Schritt die Umsetzung mit Hilfe von verschiedenen Komponenten erfolgt. Schließlich soll ein Fazit gezogen werden, ob durch die Umsetzung eine hilfreiche Erweiterung geschaffen wurde oder ob das Eintragen per Hand die bessere Methodik ist.

1 Anwendungsbereich

1.1 WordPress

Das Veranstungsvergleichsplugin ist nur in Verbindung mit dem CMS WordPress nutzbar. CMS sind Systeme, die auf die Verwaltung und Bereitstellung von Inhalten spezialisiert sind. Der Inhalt wird dabei von der Struktur, dem Aussehen und der technischen Basis getrennt. CMS eröffnen die Möglichkeit ihre Inhalte, Layouts, Designs oder Struktur flexible und schnell anpassen zu können.⁴

Der erste Vorgänger von WordPress, ein in PHP geschriebenes Weblogsystem, entstand bereits in den Jahren 2001/2002. Daraus entwickelte sich im Jahr 2004 die erste Version, die den Namen WordPress trug. Anfangs diente WordPress nur zur chronologischen Darstellung von Beiträgen. Mit der Version 1.5 kam das Verwalten von statischen Seiten als Funktion hinzu. Daraus resultierte eine erhebliche Erweiterung des CMS und erstmals die Möglichkeit konventionelle Webseiten zu erstellen.⁵

WordPress ist in zwei Teile aufspaltbar: das Backend und das Frontend. Das Backend ist der Administrationsbereich, der für Seitenbesucher normalerweise nicht sichtbar ist.⁶ Dort läuft die Verwaltung der Inhalte, Benutzer, Optionen, des Designs und der Erweiterungen ab.⁷ Das Frontend umfasst die Darstellung der Inhalte, des Designs und alle anderen Elemente, die für den Besucher sichtbar sein sollen.⁸

Mit WordPress kann eine Internetseite ohne jegliche Programmierkenntnisse aufgebaut werden. Durch Komponenten wie eine Internet-Benutzeroberfläche können Inhalte direkt erstellt und verändert werden. Anpassungen und Einstellungen sind ebenfalls über das sogenannte Dashboard aufruf- und durchführbar. Alle Inhalte werden in der MySQL-Datenbank auf dem Webserver gespeichert.⁹ Das Dashboard ist die Schaltzentrale jeder WordPress-Seite. Nach der Anmeldung auf einer WordPress-Seite gelangt der Nutzer zum Dashboard. Von hier aus können sämtliche Einstellungen getätigt und Editoren benutzt werden, um die Webseite anzupassen und zu modifizieren.¹⁰

WordPress ist durch vielfältige Möglichkeiten, u. a. im Design durch Themes oder in der Funktion durch Plugins erweiterbar. Diese Erweiterungsmöglichkeiten werden von der

⁴ Vgl. Steyer 2016, S.4-5

⁵ Vgl. Steyer 2016, S.12-13

⁶ Vgl. Steyer 2016, S.26

⁷ Vgl. Steyer 2016, S.94

⁸ Vgl. Steyer 2016, S.93

⁹ Vgl. Kommer 2019, S.7

¹⁰Vgl. Kommer 2019, S.15

Community, darunter auch Entwickler, programmiert und bereitgestellt. Neueste Webstandards wie HTML5, CSS3 und JavaScript realisieren ein responsives Design der in WordPress umgesetzten Internetseiten.¹¹

Die Grundlage für das Erscheinungsbild einer WordPress-Seite sind die Themes. Durch diese Designpakete lassen sich das Aussehen einer Internetseite ohne Programmierfähigkeiten simpel und Zeit sparend verändern. Die Trennung von Inhalt und Design spielt dabei eine wichtige Rolle. Wird etwa ein Theme durch ein anderes Theme ersetzt, bleiben die Inhalte und die Strukturen der Seite davon unabhängig in ihrem Zustand bestehen.¹²

Die technische Basis von WordPress liegt serverseitig auf PHP, hinzukommt MySQL als Datenbankverwaltungssystem. Im generierten Clientcode arbeitet WordPress mit Webtechnologien, wie clientseitigen Webframeworks. Das Hauptaugenmerk liegt dabei weiter auf Webstandards, Benutzerfreundlichkeit, simple Anpassbarkeit und Eleganz.¹³

Ein Webframework ist ein Programmiergerüst mit bestimmten Funktionalitäten, das einen Rahmen bietet, indem ein oder mehrere Programmierer eine Anwendung erstellen.¹⁴

WordPress steht als Open Source unter einer freien Lizenz. Das heißt, dass der gesamte Quelltext von jeder beliebigen Person angesehen, verändert, kopiert oder vorbereitet werden darf.¹⁵

Ein weiteres Element von WordPress sind die Custom Post Types (CPT). CPT sind neue Kategorien von WordPress-Beiträgen, die mit Hilfe der WordPress-Programmierschnittstelle erstellt werden. Diese benutzerdefinierten Elemente erscheinen in der Verwaltungsoberfläche als vollständig neue Abschnitte und können verwendet werden, um beliebige Arten von Informationen wie Veranstaltungen, Fehlerberichte, Rezepte, Filmkritiken und andere darzustellen.¹⁶

1.2 Voraussetzungen

Konzertkalender.net nutzt das Plugin „Events Manager“ zur Darstellung der Veranstaltungen. „Events Manager“ ist ein funktionsreiches WordPress-Plugin, das die Möglichkeit bietet, Veranstaltungen und Veranstaltungsorten zu veröffentlichen.¹⁷ Darüber hinaus beinhaltet das Plugin u. a. CPT, Ereigniskategorien, flexible Kalender, Frontend-Vorlagen und -

¹¹ Vgl. Steyer 2016, S.13-14

¹² Vgl. Kommer 2019, S.100

¹³ Vgl. Steyer 2016, S.12

¹⁴ Vgl. Steyer 2018, S.2

¹⁵ Vgl. Steyer 2016, S.11

¹⁶ Vgl. Lefebvre 2012, S.129-130

¹⁷ Vgl. Pixelite 2020a

Verwaltung, benutzerdefinierte Ereignis- und Standortfelder, Möglichkeiten zur Erstellung und Verwaltung von Standorten, die Integration von Google Maps, Widgets und Shortcodes zu den Funktionen¹⁸, die relevant für die Funktionalität der Seite sind. Die erstellten Veranstaltungen liefern später die benötigten Informationen für das Vergleichsplugin. Informationen, die in der Tabelle auftauchen, müssen aktuell und korrekt sein. Es ist daher immer wieder die Recherche nach aktuellen Informationen erforderlich.

1.3 Konzertkalender.net

Das Plugin findet Anwendung auf der Webseite konzertkalender.net. Eine mit dem CMS WordPress umgesetzte Internetseite rund um das Thema Musikfestivals. Neben Informationen zu Festivals und deren Veranstaltungsorten geht es um Ticketkauf und Festivalzubehör. Im Moment¹⁹ umfasst die Seite mehr als 100 Festivals und Veranstaltungsorte.

1.3.1 Aufbau der Seite

Die Webseite nutzt das WordPress-Theme: „Affiliate Theme“, das viele unterstützende Funktionen für das Affiliate-Marketing bei Internetseiten integriert hat.²⁰ Jede Seite ist gleich strukturiert und aufgebaut. Der Inhalt wird in einem Box-Layout dargestellt. Am oberen Ende der Box befindet sich die Navigationsleiste, darunter ein Slider, der verschiedene Bilder nacheinander abspielt. Den größten Teil der Seite macht die Inhaltsbox aus. Auf der rechten Seite der Box befindet sich eine Seitenleiste, die etwa ¼ der Boxbreite einnimmt. Abgeschlossen wird das Ganze von der Fußleiste. Die Farbgestaltung und einige der zuvor beschriebenen Elemente der Webseite sind in Abbildung 1 zu sehen.

¹⁸ Vgl. Pixelite 2020b

¹⁹ Stand: 11.05.2020

²⁰ Vgl. Endcore GmbH o.J.

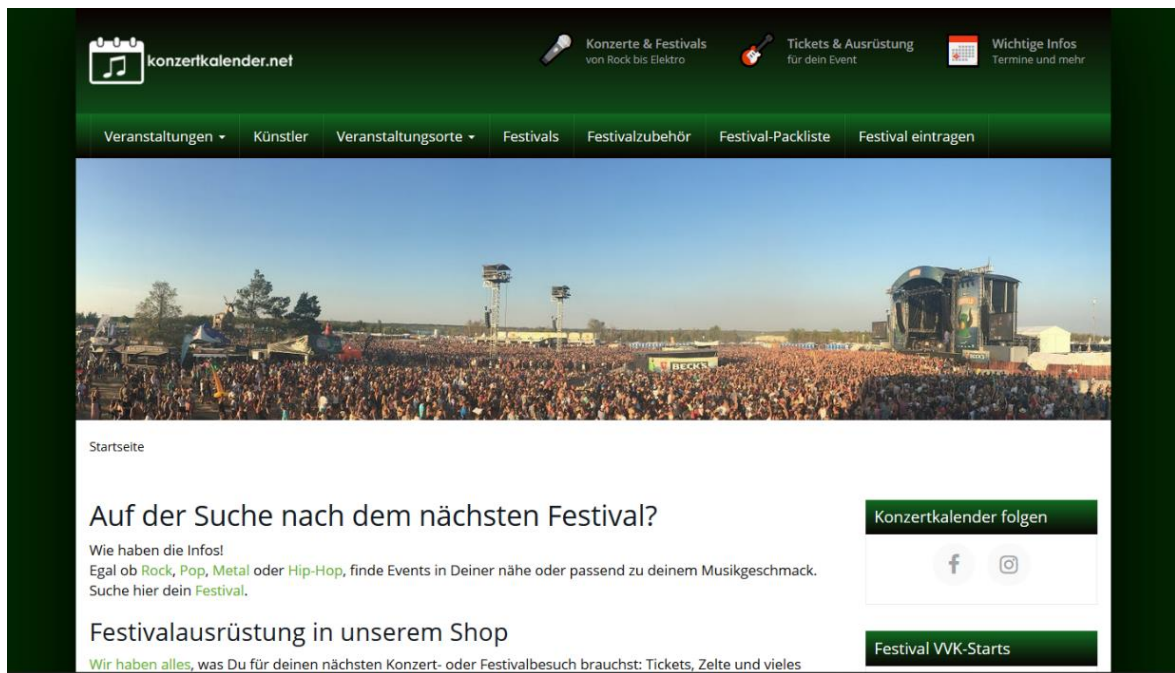


Abbildung 1: konzertkalender.net²¹

1.3.2 Aufbau einer Veranstaltung

Der wichtigste Bestandteil der Seite und die Informationsgrundlage des entstehenden Plugins sind die Veranstaltungen. Für die spezifische Formatierung als Veranstaltung in WordPress sorgt das Plugin „Events Manager“. Zum Erstellen oder Bearbeiten einer Veranstaltung sind die Informationen und Attribute im Backend eintragbar. Neben dem bei einem normalen Beitrag üblichen Editorfeld, in das zusätzliche HTML-Code, Text oder z. B. auch YouTube Videos eingefügt werden können, gibt es einen detaillierten Bereich für Einstellungen von Zeit und Datum. Zudem verfügt jede Veranstaltung über Schlagworte, die sich aus dem Band Lineup zusammensetzen. Sucht jemand auf der Seite Festivals, auf denen eine bestimmte Band oder ein bestimmter Künstler auftreten soll, werden so per Klick auf den Namen alle Auftrittsorte des Künstlers oder der Band angezeigt. Ähnlich verhält es sich mit den Veranstaltungskategorien, die die Genres und andere Eigenschaften, wie z. B. „kostenlos“ oder „outdoor“ darstellen. Das Logo des Festivals dient als Beitragsbild, falls kein Logo hochgeladen wird, nimmt ein Platzhalterbild die Stelle des Logos ein. Ergänzt werden die Informationen durch Attribute, die zusätzlich eingetragen werden können. Darunter befinden sich der Ticketpreis, eine URL zur Internetpräsenz des Festivals, ein Text, unter dem die URL schließlich verknüpft ist, Informationen zum Camping, die Zuschaueranzahl, die Anzahl der Bands und URLs für die Social-Media-Kanäle Instagram, Facebook, Twitter und YouTube. Aufgrund der momentanen Lage²² rund um das Coronavirus gibt es ein weiteres

²¹ Vgl. Konzertkalender.net 2020a

²² Stand: 11.06.2020

Attribut, welches auf Besonderheiten diesbezüglich aufmerksam macht. Der Veranstaltungsort kann über ein kleines Formular erstellt werden. Ist der Name des Ortes bereits bei Google Maps eingetragen, wird der Rest der Adressinformationen automatisch ergänzt. Später kann der Veranstaltungsort mit zusätzlichen Informationen über einen eigenen Menüpunkt erweitert werden.

Rock im Park

Konzert: Rock im Park live in Nürnberg

Abgesagt (COVID-19)

 **Datum**
05.06.2020 - 07.06.2020

Eintritt:
ab 199 €
Camping: Ja

[Jetzt Rock im Park Tickets kaufen](#)

 **Veranstaltungsort**
Zeppelinfeld Nürnberg
Zeppelinstraße
90471 Nürnberg
Deutschland

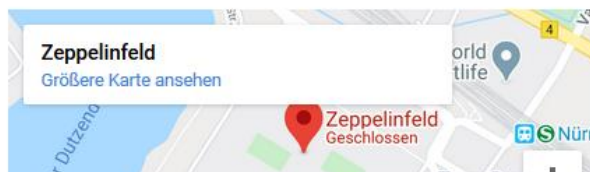


Abbildung 2: Veranstaltungsseite²³

Damit die genannten Informationen strukturiert und übersichtlich auf der Veranstaltungsseite dargestellt werden, gibt es über das Plugin „Events Manager“ im Backend eine Formatierungsseite. Mit HTML-Code kann die Seite nach Belieben formatiert werden. Durch Platzhalter werden dann im Frontend die bei den Festivals eingetragenen Attribute eingesetzt. Es entsteht eine Veranstaltungsseite wie in Abbildung 2. Neue Attribute können mit „#ATT{<Attributname>}“ im HTML-Code der Formatierungsseite eingefügt werden.

1.3.3 Funktionen

Festivals und Veranstaltungsorte

Auf der Übersichtsseite sind alle Festivals nach Datum sortiert, angefangen bei dem aktuellsten Festival. Zudem gibt es auf der Startseite eine Monatsübersicht, mit dem derzeitigen Monat, in Form eines Kalenders. An der Seitenleiste befindet sich ebenfalls eine Liste mit den aktuellen Festivals und Vorverkaufsstarts.

²³ Vgl. Konzertkalender.net 2020b

Die Veranstaltungsorte sind mit Google Maps Karten von Deutschland oder den einzelnen Bundesländern verknüpft. Rote Markierungen an den jeweiligen Veranstaltungsorten zeigen, wo sich ein Festival befindet. Eine Eingrenzung der Orte ist im Menüpunkt „Veranstaltungen nach Bundesland“ möglich. Dargestellt werden hier Veranstaltungsorte und Veranstaltungen eines ausgewählten Bundeslandes.

Suchfunktion

Das Suchfeld ist in zwei Bereiche eingeteilt. Der erste Bereich wird für einen beliebigen Suchbegriff und der Zweite für einen Ort oder eine Stadt verwendet. Die Ergebnisse beinhalten dann alle Veranstaltungen, die mit dem Suchbegriff in Verbindung stehen. Wird in den ersten Bereich z. B. „Hannover“ eingegeben, werden nicht nur die Festivals in Hannover oder die den Suchbegriff im Namen tragen angezeigt, sondern auch die, die sich in der Region Hannover befinden. Wird nach einem Künstler gesucht, werden die Festivals aufgelistet, in dessen Line-up sich der Künstler befindet.

Eine „erweiterte Suche“ bietet die Option, Suchabfragen noch weiter einzuschränken. Etwa in welchem zeitlichen Bereich, welcher Region oder Bundesland das Festival stattfinden und welches Genre gespielt werden soll. Im Zusammenspiel mit der „in der Nähe von“-Suche können durch die Angabe eines Kilometerradius die Ergebnisse räumlich genauer eingegrenzt werden.

Eintragen von Veranstaltungen

Für den Fall, dass ein Festival nicht auf der Webseite eingetragen ist, besteht die Möglichkeit dieses selbst einzutragen. Über den Menüpunkt „Festival eintragen“ wird ein Formular aufgerufen, welches dieselben Felder zum Erstellen einer Veranstaltung anbietet, die auch dem Administrator der Seite im Backend zur Verfügung stehen. Lediglich der Name und die E-Mail-Adresse müssen noch mit angegeben werden. Nach Abschicken des Formulars muss die eingetragene Veranstaltung von einem Administrator überprüft und bestätigt werden. Alle getätigten Angaben werden auf Richtigkeit kontrolliert und fehlende hinzugefügt. Hat der Administrator das neue Festival freigegeben, erscheint es auf der Internetseite. Weitergehende Pflege, wie neue Termine, Preise und anderen benötigten Informationen werden ab jetzt von den Administratoren übernommen.

Affiliate Inhalte

Konzertkalender.net hat Partnerprogramme mit Amazon und Eventim und verdient durch Affiliate-Marketing an dem Verkauf von Produkten oder Tickets. Beim Affiliate-Marketing bewirbt ein Partner eines Unternehmens deren Produkte auf seiner Internetseite. Werden

über die Seite Produkte verkauft, dann erhält der Partner eine Provision.²⁴ Bei Eventim-Tickets wird der Ticketverkauf über die Seiten der Veranstaltungen oder Veranstaltungsorte realisiert. Dort sind Weiterleitungen zum Ticketkauf für ein Festival oder einen bestimmten Ort eingebaut. Der Menüpunkt Festivalzubehör birgt einen Shop mit verschiedenen Utensilien, die für Festivalbesuche brauchbar sind. Neben Zelten und anderer Campingausrüstung werden allerlei nützliche Sachen zum Thema Festival angeboten. Alle Artikel können über einen Button, der zur Amazon-Webseite weiterleitet, gekauft werden. Auf der Seitenleiste ist eine Übersicht mit den neusten Produkten filterbar mit Hilfe eines Preisspannenreglers für bestimmte Kategorien angeordnet. Kauft ein Nutzer etwas über einen Amazon- oder Eventim-Link, verdient konzertkalender.net einen Anteil am Verkauf des Produktes oder Tickets.

Suchmaschinenoptimierung

Um im Internet besser aufgefunden zu werden benutzt konzertkalender.net Werkzeuge zur Suchmaschinenoptimierung (SEO). Eines davon ist die schema.org-Auszeichnung. Schema.org ist eine Initiative, welche von Google, Microsoft, Yahoo und Yandex mit dem Ziel gegründet wurde, Schemata für strukturierte Daten auf Webseiten im Internet zu erstellen, zu pflegen und zu fördern. Hierzu können verschiedene Codierungen verwendet werden.²⁵

Die Veranstaltungen auf konzertkalender.net werden mit schema.org als Musikevent deklariert. Ebenfalls werden Datum, Beschreibung sowie der Ort ausgezeichnet und im optimalen Fall in den Ergebnissen der Google-Suche angezeigt. Als Beispiel dafür dient die Abbildung 3, die die Suche nach dem Festival „Hunteburger Open Air Festival“ zeigt. Im Ergebnis von Google wird auf der rechten Seite ein Ereignis angezeigt. Beim Überprüfen der Quelle wird ein der Link zu der Veranstaltungsseite des Festivals auf konzertkalender.net angezeigt.

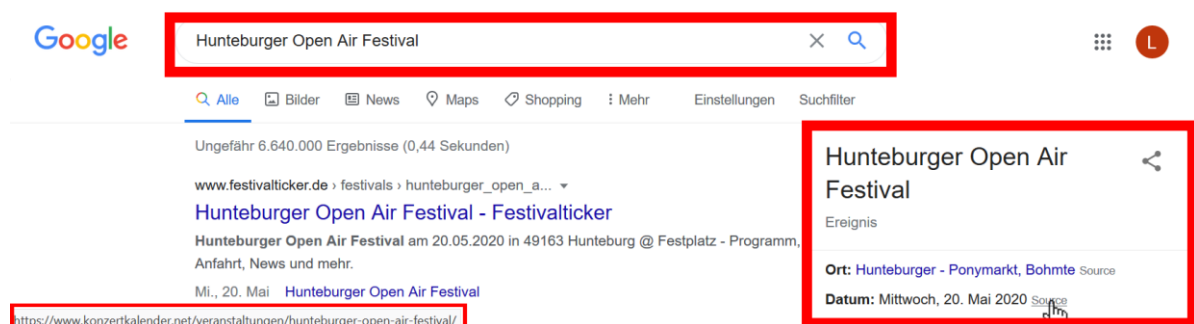


Abbildung 3: Beispiel Schema.org (Google)²⁶

²⁴ Vgl. Lammenett 2019, S.63-64

²⁵ Vgl. Schema.org o.J.

²⁶ Vgl. Google 2020. (Bild durch Autor bearbeitet)

Die Schema.org Auszeichnung ist per HTML-Code in den Veranstaltungsseite implementiert. Der Code wird mithilfe des Plugins „Events Manager“ in die Formatierung jeder Seite eingebaut.

Eine weitere Hilfe zur SEO ist das Plugin „Yoast SEO“. Es hilft u. a. bei der Optimierung von Schlüsselwörtern, Schlüsselsätzen, Synonymen und der Anpassung von Snippets.²⁷

Snippets sind Beschreibungen und Darstellungen der einzelnen Treffer einer Suchmaschine im Internet. Basisinformationen eines Snippets sind Überschrift, Beschreibungstext und ein Uniform Resource Locator (URL). Das Snippet repräsentiert lediglich das Dokument dahinter und ist eine Interpretation durch die Suchmaschine.²⁸

²⁷ Vgl. Yoast BV 2020

²⁸ Vgl. Lewandowski 2018, S.149-150

2 Problemstellung

Das größte Problem beim Erstellen eines Vergleiches von Festivals, auf einer Internetseite, ist wohl der Aufwand, der für das Eintragen betrieben werden muss. In Bezug auf den Anwendungsbereich konzertkalender.net, müssten die bereits auf der Seite eingetragenen Veranstaltungen noch ein weiteres Mal erfasst werden. Bei der Anwendung eines fertigen Plugins zur Generierung von Tabellen gibt es keine Möglichkeit die Veranstaltungen in die neue Tabelle zu kopieren. In dem Fall müsste alles per Hand eingetragen werden. Eine Lösung ohne Plugins nur mit HTML-Code wäre möglich, aber stieße ebenfalls wieder auf das Problem der redundanten Informationsangabe. Hinzu kommt eine gewisse Unübersichtlichkeit. Bei mehr als 100 Veranstaltungen, die derzeit auf der Seite einzusehen sind, die Übersicht zu behalten und Fehler zu vermeiden oder zu korrigieren ist umständlich und unkomfortabel. Bei beiden Varianten muss außerdem bedacht werden, dass Veranstaltungen sich von Jahr zu Jahr verändern und permanent sowohl die betreffenden Informationen, als auch die Übersicht in der Tabelle aktuell gehalten werden müssen, was regelmäßig einen zusätzlichen Aufwand zur Folge hat.

Das Plugin soll also möglichst komfortabel für den Nutzer die Informationen aus den bereits vorhandenen Veranstaltungen extrahiert, in eine Tabelle einfügen und diese schnell und einfach aktualisierbar machen. Es sollte dabei keine zusätzliche Dateneingabe nötig sein.

3 Konzept

3.1 Aufbau

3.1.1 Frontend

Auf der Seite konzertkalender.net für die das Plugin konzipiert wird, wird der Inhalt auf einem weißen Hintergrund abgebildet. Normalerweise befindet sich auf der rechten Seite eine Seitenleiste mit Informationen. Um mehr Platz für die abgebildete Tabelle zu haben, wird diese im Fall der Vergleichsseite aber ausgeblendet. Die durch das Plugin generierte Seite kann somit die ganze Breite zur Darstellung des Inhalts nutzen.

Die Tabelle wird in den Zeilen die Veranstaltungen und in den Spalten die Informationen für den Vergleich anzeigen. Zunächst sind bis zu acht Vergleichszeilen vorgesehen. Zu einem späteren Zeitpunkt könnten diese, dadurch, dass die Veranstaltungen untereinander gelistet werden, noch um beliebig viele weitere Zeilen ergänzt werden. Der Seitenaufbau ermöglicht eine Ausdehnung der Tabelle nach unten.

In Anwendung auf die Seite konzertkalender.net werden in den Spalten die zu vergleichenden Attribute des Festivals aufgeführt. Insgesamt gibt es acht Spalten. In Abbildung 4 sind die geplanten Vergleichskriterien abgebildet. Darüber hinaus befindet sich in der Festival-spalte in jeder Zeile ein Auswahlmenü, aus dem das gewünschte Festival ausgewählt werden kann. Die Sortierung der Festivals erfolgt alphabetisch.

Festival	Beginn	Ende	Tage	Preis (€)	Camping	Zuschauer	Anzahl Bands
Festival 1	10.10.2020	11.10.2020	2	22	Ja	3.000	5
Festival 2	04.06.2020	06.06.2020	3	67	Ja	80.000	40
Festival 3	13.09.2020	13.09.2020	1	20	Nein	20.000	4
Festival 4	22.07.2020	25.07.2020	4	100	Ja	66.000	80
Festival 5	01.09.2020	01.09.2020	1	18	Nein	200	5
Festival 6	08.04.2020	11.04.2020	4	120	Ja	50.000	44
Festival 7	19.05.2020	19.05.2020	1	15	Nein	2.000	10
Festival 8	22.00.2020	25.03.2020	4	111	Ja	56.000	70

Abbildung 4: Entwurf der Tabelle ²⁹

3.1.2 Menü

Im Menü des WordPress-Backends wird das Plugin einen eigenen Menüpunkt besitzen. Die Abbildung 5 gibt eine Übersicht über den geplanten Aufbau des Menüs. Beim Anklicken dieses Punktes erscheinen die Funktionen als Unterpunkte. Über Anklicken des Menüpunktes oder der einzelnen Unterpunkte, werden die Seiten zu den entsprechenden Funktionen

²⁹ Eigene Darstellung

aufgerufen. Unterpunkte sind hier das Importieren von bereits auf der Seite veröffentlichten Veranstaltungen, das Einfügen von Festivals per Hand, das Löschen von allen eingetragenen Festivals und eine Seite zum Shortcode. Als Logo für das im WordPress-Menü angezeigte Plugin, dient das Logo der Seite konzertkalender.net.

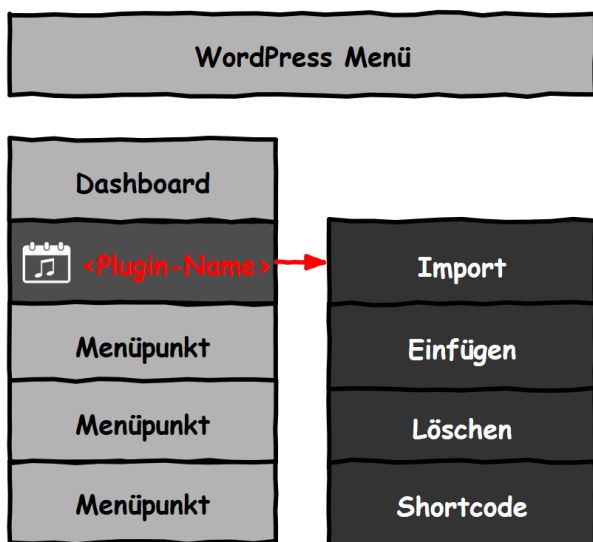


Abbildung 5: Menü Entwurf³⁰

3.1.3 Installierte Plugins

Im Dashboard gibt der Menüpunkt „Plugins“ eine Übersicht, welche Plugins bereits installiert sind. Hier wird nach der erfolgreichen Installation auch das hier behandelte Plugin aufgelistet. Es werden dabei Informationen zum Autor sowie die Version des Plugins und ein beschreibender Text angezeigt. Darüber hinaus sind hier auch die Optionen zum Aktivieren bzw. Deaktivieren oder Deinstallieren des Plugins zu finden.

3.2 Funktionen

3.2.1 Import

Der Import soll über einen Button im Backend ausführbar sein. Nach einem Klick auf den Button werden die bereits vorhanden Veranstaltungen von der Seite ausgelesen. Danach erscheint eine Liste mit den vorhandenen Einträgen. Es besteht die Möglichkeit alle importierten Veranstaltungen wieder zu löschen oder einzelne Veranstaltungen zu entfernen. Im Frontend werden nach dem Import alle in WordPress gespeicherten Veranstaltungen in der Tabelle sichtbar.

3.2.2 Einfügen

Für den Fall, dass eine Veranstaltung nicht auf der Seite aufgeführt werden, aber in der Tabelle erscheinen soll, kann diese per Hand eingepflegt werden. Dafür steht ein Formular im

³⁰ Eigene Darstellung

Backend bereit, welches alle nötigen Felder für die relevanten Informationen bietet. Beim Klicken auf den „Einfügen“-Button erscheint die erstellte Veranstaltung im Backend in einer Liste aller selbst erstellter Elemente. Im Frontend ist die Veranstaltung in der Tabelle auswählbar. Es besteht auch hier die Möglichkeit alle über das Formular erstellten Veranstaltungen wieder zu löschen oder einzelne Veranstaltungen zu entfernen.

3.2.3 Löschen

Unter den Menüpunkten für den Import und das Einfügen können einzelne der ausgewählten Funktion gelöscht werden. Auch ein komplettes Löschen aller Elemente ist möglich. Dies geschieht über den Menüpunkt „Löschen“.

3.2.4 Shortcode

Der Shortcode dient dem Einfügen der Vergleichstabelle auf Seiten der WordPress-Installation oder Beiträgen der Seite. Die Tabelle wird mit einem Shortcode der folgenden Form aufrufbar sein: [Shortcode].

4 Komponenten

4.1 Hypertext Markup Language

Im Jahr 1989 wurde die Hypertext Markup Language (HTML) vom britischen Informatiker Tim Burners-Lee veröffentlicht. Burners Entwicklung hatte die Beschreibung und Verlinkung von Texten zur Aufgabe. HTML codiert alle Bestandteile einer Webseite in einem weltweit gültigen Standard. Dabei sind die Dateien reine Textdateien mit einer geringen Datenmenge. Medien wie Bilder, Grafiken oder Videos können ergänzt werden, hierfür müssen diese jedoch auf dem Endgerät vorliegen. HTML ist eine Sprache, um Hypertext zu notieren. Dieser Hypertext ist ein wesentlicher Bestandteil des World Wide Webs (WWW), da Texte mithilfe von Links miteinander verbunden werden können. Dementsprechend verlassen Hypertexte lineare Strukturen von gedruckten Texten und erlauben das Hin- und Herspringen innerhalb dieser Texte. Befinden sich verbundene Textdateien auf verschiedenen mit dem Internet verbunden Endgeräten, entsteht ein Informationssystem. Resultierend aus diesen Entwicklungen entstand das WWW.³¹

4.2 Cascading Style Sheets

Stylesheets sind ein wichtiger Teil des heutigen Webdesigns. Die Cascading Style Sheets (CSS) wurde 1996 als Gestaltungssprache im Internet standardisiert. CSS erweitert die Beschreibungssprache HTML um die Fähigkeit, Formatierungen an Texten und Elementen vorzunehmen. Gestaltungsoptionen eines einzelnen Elementes, Absatzes oder des gesamten Dokumentes sind beispielsweise Schriftart und -größe, Breite von Absätzen und Farbe von Texten und Hintergründen. Die Dokumentenformatierung kann in separate Dateien ausgelagert werden und benötigen meist nur wenig Code. Die daraus resultierenden kleinen Dateigrößen begünstigen eine kurze Ladezeit zur Folge.³² Durch das Einbinden von einem Stylesheet, das als Formatvorlagen für logisch strukturierte Dokumente wie HTML dient, wird eine Trennung von Inhalt und Gestaltung vollzogen. Bei einer konsequenten Anwendung der gültigen Standards wird ein einheitliches Layout sichergestellt, sodass Änderungen, die auf mehrere Seiten Auswirkungen haben sollen, nur an einer Stelle bearbeitet werden müssen. Mit CSS lässt sich die Kompatibilität bei Nutzung verschiedener Bildschirme ermöglichen und problemlos an Smartphones, Tablets, Laptops oder auch hochauflösenden Bildschirmen anpassen.³³

³¹ Vgl. Bühler u. a. 2017, S.2-3

³² Vgl. Schröder 2012, S.16-17

³³ Vgl. Haunschild 2017, S.8-9

4.3 PHP: Hypertext Preprocessor

Ursprünglich hieß die Open-Source-Skriptsprache PHP „Personal Home Page Tools“. Heutzutage ist „PHP: Hypertext Preprocessor“ (PHP) die gängige Bezeichnung. Durch PHP können auf Internetseiten Interaktionen eingebaut, Datenbanken gesteuert, sowie die Webseite individuell an Benutzerverhalten angepasst werden. Einfacher ausgedrückt eröffnet PHP Möglichkeiten, die mit HTML schlichtweg nicht umsetzbar sind.³⁴ Aufgrund des großen Funktionsumfangs wird PHP für die Programmierung dynamischer Webseiten und Internetanwendungen verwendet. Bei der Anwendung werden PHP-Anweisungen direkt auf dem Server ausgeführt und nicht vom Client. Diese Anweisungen sind für Betrachtende nicht im Quellcode ersichtlich, sondern lediglich für den Webserver wahrnehmbar.³⁵

In Abbildung 6 ist die Funktionsweise von PHP dargestellt. Um die Anweisungen auf der Internetseite zu visualisieren, sendet der Client per Hypertext Transfer Protocol Secure (HTTPS) eine Anfrage nach einer PHP-Datei an den Webserver. Ist die Datei als PHP-Request identifiziert, wird sie im Dateisystem gesucht und zur Verarbeitung an den PHP-Interpreter übergeben. Nachdem der Interpreter den Code zu einer ausführbaren Form verarbeitet hat, sendet er diese zurück zum Webserver. Der Client bekommt eine Antwort vom Webserver und als Ergebnis entsteht ein dynamisch erzeugtes HTML-Dokument mit der Dateierendung „.php“.³⁶

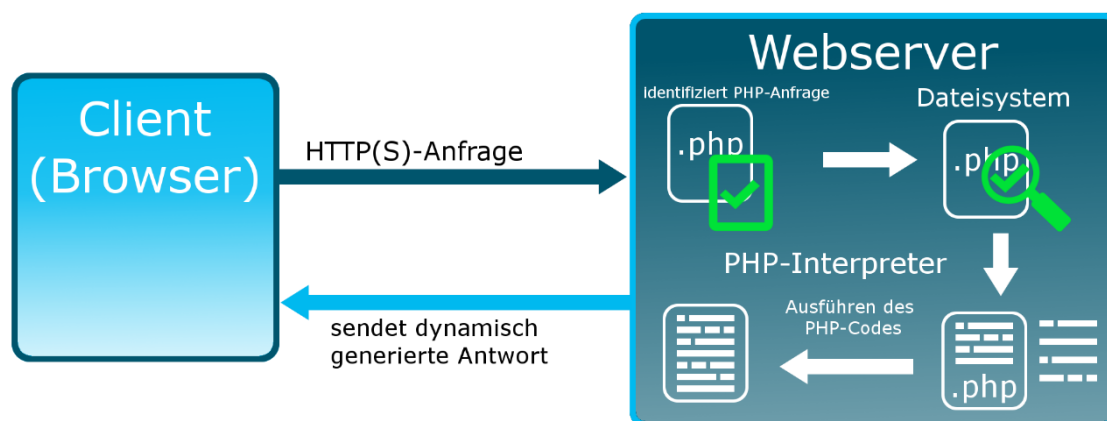


Abbildung 6: Funktionsweise PHP³⁷

4.4 Structured Query Language

Structured Query Language (SQL) wird zur Bearbeitung und Auswertung von relationalen Datenbanken genutzt. SQL gilt als deklarative Sprache, weil die angegebenen Befehle und

³⁴ Vgl. Heller 2016, S.7

³⁵ Vgl. Heller 2016, S.10

³⁶ Vgl. Steyer 2019, S.11

³⁷ Eigene Darstellung vgl. Steyer 2019 S.11

nicht die Art und Weise bestimmt, welches Ergebnis angezeigt wird. Hierbei wird kein Algorithmus implementiert, sondern der Datenbankserver ermittelt das Ergebnis, das durch den SQL-Befehl gefordert wird.³⁸ Mehrere Anweisungen lassen sich über sogenannte SQL-Skripts zusammenfassen und automatisiert ausführen. SQL besteht aus vier, in Abbildung 7 ersichtlichen, Hauptbestandteilen. Neben den Bereichen der Data Query Language (DQL), welches das Abfragen von Daten ermöglicht und der Data Definition Language (DDL) zum Erstellen von Datenbanken, Relationen und Indizes, umfasst der Sprachumfang von SQL noch zwei weitere Befehlsgruppen. Zum einen die Data Manipulation Language (DML), zum Anlegen, Ändern und Löschen von Datensätzen und zum anderen die Data Control Language (DCL), bei der es größtenteils um das Anlegen von Benutzern und Vergabe von Zugriffsrechten geht. Zusammengefasst dient SQL als Zugriff und Anwendung zur Kommunikation mit einer Datenbank durch Eingabe von Anweisungen.³⁹

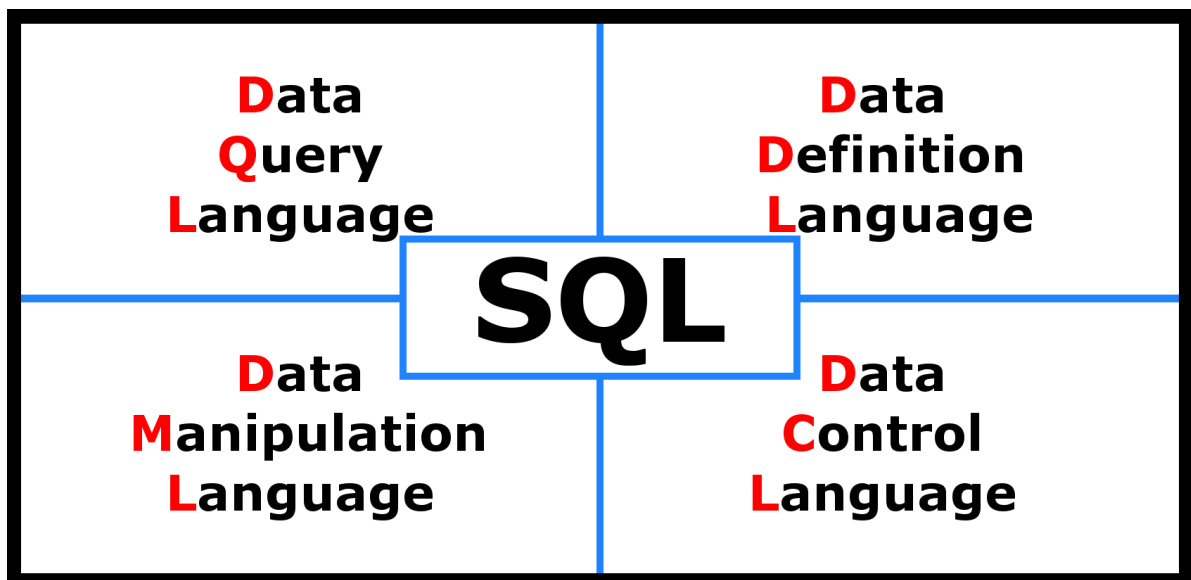


Abbildung 7: Sprachumfang von Befehlsgruppen in SQL⁴⁰

SQL entwickelte sich nach der ersten Normierung im Jahr 1987 zur wichtigsten Standardsprache für Datenbanken. Im Laufe der Geschichte wurde die Zugriffssprache immer wieder erheblich erweitert und zahlreiche Ergänzungen vorgenommen.⁴¹

³⁸ Vgl. Adams 2020, S.63

³⁹ Vgl. Fuchs 2018, S.68

⁴⁰ Eigenen Darstellung vgl. Fuchs 2018, S.68

⁴¹ Vgl. Schicker 2017, S.99-100

5 Implementierung

5.1 Aufbau

Der Aufbau des Plugins ist in Abbildung 8 dargestellt. Dabei basiert es auf der `index.php` (5.2.1), in der das Grundgerüst des Plugins definiert ist. Die `index`-Datei inkludiert verschiedene Dateien und Funktionen, um dem Plugin letztendlich seine Funktionalität geben zu können. Die Startseite, die Importfunktion sowie die Erzeugung der Tabelle werden von der `index.php` eingebunden. Für die Funktionalität brauchen die Importfunktion und die Tabelle jedoch noch zusätzliche Dateien. Außer der benötigten Verbindung zur Datenbank werden weiterhin Dateien und Funktionen zur Erzeugung einer neuen Tabelle im Datenbanksystem, eine Ausgabe von importierten Elementen, die Berechnung des Abstandes zweier Tage und die Umwandlung des Datums in ein anderes Format gebraucht. Neben der `index.php` befindet sich im Hauptverzeichnis des Plugins die `uninstall.php` (5.2.2) und die Unterordner für CSS-Dateien, Bilder und inkludierte PHP-Dateien.

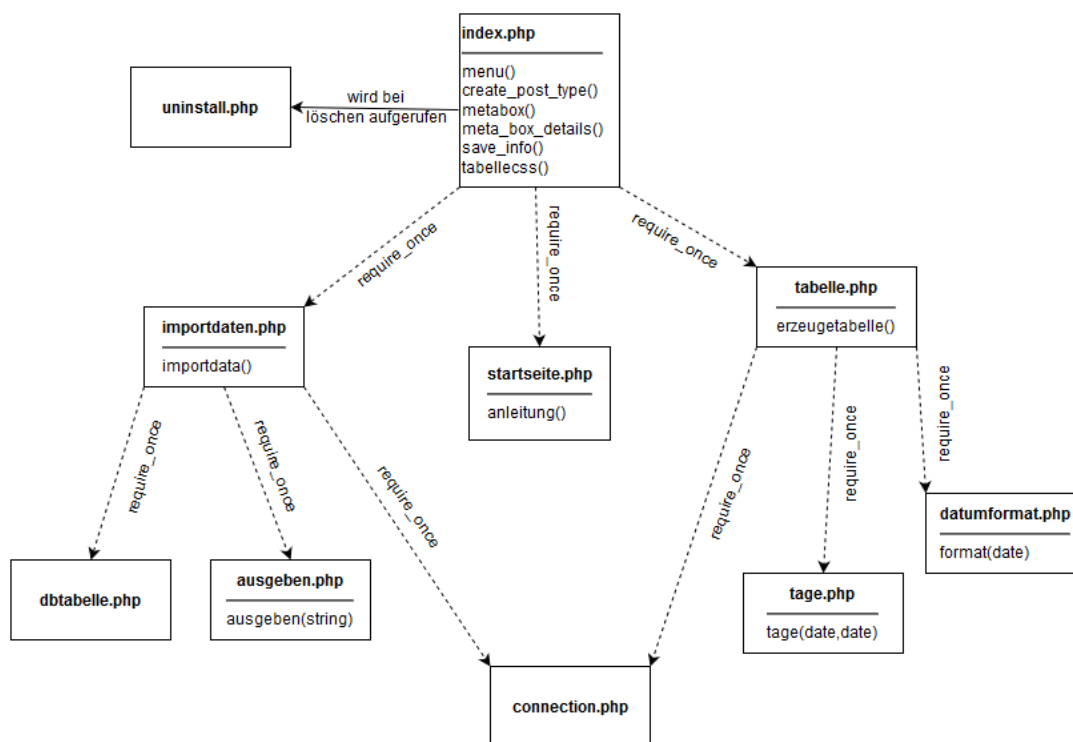


Abbildung 8: Aufbau Plugin⁴²

5.2 Dateien

Im Folgenden werden die einzelnen Dateien des Plugins, in ihrer Funktion und ihrem Zusammenhang, genauer erläutert. PHP-Funktionen⁴³ ab Abschnitt 5.2.3 bis zum Ende des 5.

⁴² Eigene Darstellung

⁴³ Im Text *kursiv*-gekennzeichnet

Kapitels werden anhand eines PHP-Manuals erklärt.⁴⁴ Zudem kommt es zur Nennung von Funktionen, Tabellen, Variablen und weiteren Elementen, die bei der Programmierung des Vergleichsplugin verwendet wurden.⁴⁵

5.2.1 index.php

Kopfzeilenkommentar

Die index.php (Anhang 1: index.php) ist die zentrale Datei des Plugins. Eine Plugin-Datei beginnt mit einem Kopfzeilenkommentar. Der Quellcode 1 zeigt das Kopfzeilenkommentar. Im Gegensatz zu einer normalen PHP-Datei, in der die Kommentare rein optional sind, ist in WordPress-Plugin- und Theme-Dateien der Information-Header erforderlich. Die Kommentarzeilen, die u. a. Informationen zum Autor, der Version und eine Beschreibung des Plugins liefern, helfen WordPress, die Datei als Plugin zu identifizieren. Fehlen diese Kommentare, dann taucht das Plugin nicht im WordPress-Plugin-Menü auf und kann so auch nicht genutzt werden.⁴⁶ Im Falle des Veranstungsvergleichsplugins wird im Kopfzeilenkommentar zuerst der Name des Plugins, ohne Link zur Plugin-Seite aufgeführt, da diese nicht vorhanden ist. Hinzu kommt eine kurze Beschreibung, die Version und der Name des Autors mit seinem persönlichen Seitenlink.

Des Weiteren werden die für Funktionalität benötigten Dateien importdaten.php (5.2.4), tabelle.php (5.2.7) und startseite.php (5.2.11) eingebunden.

Quellcode 1: Plugin-Kopfzeilenkommentar⁴⁷

```
/*
Plugin Name: WVP-Veranstungsvergleichsplugin
Plugin URI:
Description: Ein Plugin zum Vergleich von Veranstaltungen.
             Importiere bereits vorhanden Veranstaltungen aus
             dem Events Manager, um sie miteinander zu vergleichen oder
             füge eigene Veranstaltungen der Tabelle hinzu.
Version: 1.0
Author: Lukas Fischer
Author URI: https://www.konzertkalender.net/
*/
```

⁴⁴ Das Manual findet sich unter www.php.net/manual/de/index.php [Abruf am 04.06.2020].

⁴⁵ In “...”-gekennzeichnet

⁴⁶ Vgl. Bondari 2011, S.30-31

⁴⁷ Basiert auf vgl. Lefebvre 2012, S.41

Hooks

Der Inhalt der index.php besteht aus Wordpress-Hooks und den dazugehörigen Funktionen. Hooks sind die Aktivierungsfunktionalität, die Plugins in WordPress erst möglich machen. Ein Hook ermöglicht die Ausführung von zusätzlichem Code an bestimmten Stellen. Dieser Code fügt in der Regel Inhalt zu einer Seite hinzu oder ändert die Art und Weise, wie eine bestimmte Aktion ausgeführt wird. Jedes Plugin kann eine oder mehrere Funktionen mit einem Action-Hook registrieren. Die Funktion "add_action" wird dabei verwendet, um benutzerdefinierten Plugin-Code mit einem der WordPress-Hooks zu verknüpfen. An "add_action" müssen zwei Parameter zwingend und zwei weitere optional übergeben werden. Der erste Parameter gibt den Namen des gewünschten WordPress-Hooks an. Parameter Nummer zwei ist der Name der Plugin-Funktion, die aufgerufen wird, um eine Aktion durchzuführen. Die Prioritätsparameter sind optional. Sie geben die Ausführungspriorität dieses Plugins, relativ zu anderen Plugins an.⁴⁸

Menüpunkt

Um die wichtigsten Funktionen übersichtlich zu trennen, hat das Plugin ein Menü mit zwei Unterseiten. Das Menü und die Unterseiten werden per Hook registriert. Die Funktion, die durch den Hook registriert wurde, enthält Unterfunktionen mit dem Namen "add_menu_page" und "add_submenu_page". Sie sorgen dafür, dass im Backend der Menüpunkt mit den Untermenüpunkten erscheint. Hierfür werden jedoch einige Parameter benötigt.

Die ersten beiden Parameter der Funktion "add_menu_page" sind Zeichenketten und dienen als Titel der Seite und als Menüpunkttext. Des Weiteren werden Benutzerrechte, der "menu_slug" und die aufzurufende Funktion mit angegeben.⁴⁹ Hinzu kommt noch ein optionaler Parameter, der es erlaubt, ein benutzerdefiniertes Icon im Menü neben dem neuen Hauptmenüpunkt anzuzeigen. Nachdem der erste Hauptmenüpunkt erstellt wurde, kann die Funktion "add_submenu_page" verwendet werden, um die Untermenüpunkte anzuhängen. Die angegebenen Parameter sind nicht ganz identisch mit dem Hauptmenüpunkt. Der erste Parameter hier, ist der "menu_slug" des Hauptmenüs.⁵⁰

Durch "add_action" und den Parameter "admin_menu" wird dem Vergleichsplugin ein Menü hinzugefügt. Zwei Untermenüpunkte, der eine zum Importieren der Inhalte, welcher die "importdata"-Funktion aufruft, und der andere, der die Seite zum Erstellen der eigenen

⁴⁸ Vgl. Lefebvre 2012, S.46-47

⁴⁹ Vgl. Lefebvre 2012, S.84

⁵⁰ Vgl. Lefebvre 2012, S.87

Einträge aufruft, werden benötigt. Beim Klick auf den Hauptmenüpunkt wird die “anleitung“-Funktion aufgerufen. Im Quellcode 2 wird zuvor beschriebenes umgesetzt.

Quellcode 2: Menü einfügen⁵¹

```
(...)  
add_action('admin_menu', 'menu');  
function menu(){  
    add_menu_page( 'Festivalvergleich',  
                  'Vergleichstabelle',  
                  'manage_options',  
                  'vergleich_plugin',  
                  'anleitung',  
                  plugins_url( 'images/konzertkalender.png', __FILE__ )  
    );  
    add_submenu_page('vergleich_plugin',  
                    'Importfunktion',  
                    'Import',  
                    'manage_options',  
                    'vergleich_plugin_import',  
                    'importdata' );  
(...)
```

Custom Post Type

Für ein einfaches Einfügen und Löschen von eigenen Veranstaltungen, hat der Nutzer die Möglichkeit, über einen CPT eine eigene Veranstaltung zu kreieren. Wie ein CPT hinzugefügt wird, zeigt Quellcode 3.

Um einen neuen CPT hinzuzufügen, wird über die “add_action“-Funktion der Hook “init“ und ein Funktionsname aufgerufen.⁵² In der Funktion wird durch den Aufruf von “register_post_type“ WordPress über die Existenz eines neuen Posttyps informiert. Dabei sind zwei Argumente erforderlich: der Namen des Posttyps und ein Array mit einer Reihe von Parametern, die als Eigenschaften dienen.⁵³

Der im Plugin angelegte CPT heißt “veranstaltung“. Der Parameter für die Eigenschaften des CPTs ist ein Array, welches im ersten Element ein weiteres Array beinhaltet. Der Inhalt des Arrays beschreibt Elemente, wie z. B. neuer Eintrag oder Eintrag bearbeiten, die beim

⁵¹ Basiert auf vgl. Lefebvre 2012, S.85-87

⁵² Vgl. Lefebvre 2012, S.131

⁵³ Vgl. Lefebvre 2012, S.133

Aufruf der CPT-Seite im Backend angezeigt werden. Die Eigenschaften außerhalb des Arrays steuern: die Sichtbar, die Auffindbarkeit in der Frontend-Suche, die Position im Admin-Menü angezeigt werden soll und welche Elemente bei der Erstellung bzw. Bearbeitung vorhanden sein sollen.⁵⁴ Beim “supports“-Attribut wird im Falle des Plugins nur der Titel angezeigt. Elemente wie Kommentare oder der Editor werden ausgeblendet.

Quellcode 3: Hinzufügen eines CPT⁵⁵

```
(...)  
add_action( 'init', 'create_post_type' );  
function create_post_type() {  
    register_post_type( 'veranstaltung',  
        array(  
            'labels' => array(  
                'name' => 'Ein&uuml;gen',  
(...)  
            ),  
            'public' => true, //  
            'supports' => array( 'title' ),  
            'exclude_from_search' => true,  
            'show_in_menu' => 'edit.php?post_type=veranstaltung',  
        )  
    );  
}(...)
```

Metabox

Dem CPT fehlen jetzt noch alle Elemente, außer dem Titel. Für neue Elemente wird eine Metabox für das CPT hinzugefügt.

In WordPress gibt es ein eingebautes Meta-Box-System, um eine Schnittstelle zu erstellen, die es Benutzern ermöglicht für CPT spezifische Felder anzulegen. Der Quellcode 4 bildet das Hinzufügen einer Metabox ab. Meta-Boxen werden mit der Funktion “add_meta_box“ erstellt. Zusätzlich zur Deklaration der Meta-Box und ihrer Zuordnung zum CPT, wird ein Callback für das Rendern des Inhalts der Box definiert. Die Funktion “add_meta_box“ enthält eine weitere Funktion, die den Inhalt der Meta-Box wiedergibt. Sobald die enthaltenen Daten aus der Datenbank abgerufen wurden, können sie verwendet werden, um die neuen Felder auf dem Bildschirm darzustellen.

⁵⁴ Vgl. WordPress.org o. J.a

⁵⁵ Basiert auf vgl. Lefebvre 2012, S.130-134

Zuletzt wird eine Funktion registriert, die aufgerufen wird, wenn der neue CPT vom Administrator gespeichert oder gelöscht wird. Dabei muss zuerst der empfangene CPT überprüft werden. Wenn es sich um den entsprechenden Typen handelt, wird geprüft, ob Postdaten eingegangen sind und dementsprechend gespeichert oder gelöscht.⁵⁶

Für das CPT beim Vergleichsplugin wird die Metabox über “add_action“ mit dem Parameter “admin_init“ und einer Funktion registriert. Der Metabox wird die Funktion “meta_box_details“ hinzugefügt, die später die für den Vergleich benötigten Elemente definiert. Beim Erstellen einer neuen Veranstaltung durch den CPT, erscheinen in der Metabox Eingabefelder für die benötigten Details, wie das Start- und Enddatum.

Quellcode 4: Meta-Box hinzufügen⁵⁷

```
(...)  
add_action( 'admin_init', 'metabox' );  
function metabox() {  
    add_meta_box( 'details_meta_box', 'Informationen',  
        'meta_box_details', 'veranstaltung', 'normal', 'high' );  
}  
function meta_box_details( $veranstaltung ) {  
    $start = esc_html( get_post_meta( $veranstaltung->ID, 'beginn', true ) );  
    $ende = esc_html( get_post_meta( $veranstaltung->ID, 'ende', true ) );  
    (...)  
    <tr>  
        <td style="width: 100%">Von</td>  
        <td><input type="date" size="80" name="startdatum" value="<?php  
            echo $start; ?>" /></td>  
    </tr>  
    (...)
```

Shortcode

Die Vergleichstabelle wird per Shortcode aufgerufen. Ein Shortcode wird im Text eines Beitrags oder einer Seite von einem Paar eckigen Klammern umgeben. Dieser wird von der WordPress-Engine identifiziert, die dann nach Funktionen sucht, die mit diesem speziellen Code verknüpft sind. Wenn sie gefunden werden, wird die zugehörige Funktion aufgerufen

⁵⁶ Vgl. Lefebvre 2012, S.137-138

⁵⁷ Basiert auf vgl. Lefebvre 2012, S.135-138

und das Ergebnis verwendet, um den ursprünglichen Shortcode-Text im Inhalt des Artikels zu ersetzen.⁵⁸

In Quellcode 5 ist der Programmiercode zum Hinzufügen eines Shortcodes dargestellt. Um das Plugin mit der Funktionalität eines Shortcodes zu erweitern, wird in der `index.php` die Funktion `add_shortcode` ergänzt. Der erste Parameter ist der Ausdruck, der in den eckigen Klammern stehen muss, damit die Funktion, der zweite Parameter, aufgerufen wird. Um die Vergleichstabelle im Frontend abbilden zu lassen, muss der Shortcode `„[vergleichstabelle]“` per WordPress-Editor in einer Seite oder einem Beitrag gespeichert sein. Hierbei wird dann die Funktion `erzeugetabelle` aus der `tabelle.php` (5.2.7) aufgerufen.

*Quellcode 5: Shortcode hinzufügen*⁵⁹

```
(...)  
add_shortcode( 'vergleichstabelle', 'erzeugetabelle' );  
(...)
```

CSS

Damit die vom Plugin im Frontend generierte Tabelle ansehnlich und für die mobile Nutzung brauchbar ist, muss die `tabelle.css` eingebunden werden.

Durch den Hook `template_redirect` wird die dazugehörige Funktion ausgeführt, bevor WordPress bestimmt, welche Vorlagenseite geladen werden soll.⁶⁰ In der übergebenen Funktion wird `wp_enqueue_style` aufgerufen, die eine angegebene CSS-Datei anwendet⁶¹, wenn die Seite mit dem Vergleichsplugin aufgerufen wird. Quellcode 6 zeigt das Registrieren eines Hooks zum Einfügen einer CSS-Datei.

*Quellcode 6: Plugin CSS hinzufügen*⁶²

```
add_action( 'template_redirect', 'tabellecss' );  
function tabellecss() {  
    if ( is_page( 'vergleich' ) ) {  
        wp_enqueue_style( 'style-name',  
            plugins_url( '/css/tabelle.css', __FILE__ ) );  
    }  
}
```

⁵⁸ Vgl. Lefebvre 2012, S.64

⁵⁹ Basiert auf vgl. Lefebvre 2012, S.63-65

⁶⁰ Vgl. WordPress.org o. J.b

⁶¹ Vgl. WordPress.org o. J.c

⁶² Vgl. Anhang 1: `index.php`

5.2.2 uninstall.php

Wurde ein Plugin über das Plugin-Menü im Dashboard deaktiviert, kann der Administrator dieses per Klick löschen. Beim Löschen überprüft WordPress, ob die Datei `uninstall.php` (Anhang 2: `uninstall.php`) vorhanden ist. Existiert im Plugin-Verzeichnis eine Datei mit diesem Namen, dann startet WordPress das Löschen der Plugin-Dateien. Dabei wird geprüft, ob eine bestimmte Konstante vorhanden ist, die WordPress vor Ausführen des Skripts gesetzt haben sollte. Falls diese nicht vorhanden ist, wird das Skript aus Sicherheitsgründen abgebrochen. Anderenfalls wird der weitere Inhalt, der in Quellcode 7 zu sehen ist, ausgeführt.⁶³

Bei dem hier konstruierten Vergleichsplugin werden nach Überprüfung der Konstanten neben den Dateien des Plugins, auch Elemente der Datenbank, die vom Plugin erstellt worden sind, gelöscht. Es handelt es sich hierbei um die Tabelle "vergleich" und die Einträge des CPT "veranstaltung".

*Quellcode 7: `uninstall.php`*⁶⁴

```
if ( !defined( 'WP_UNINSTALL_PLUGIN' ) ) {
    exit();
}
global $wpdb;
$wpdb->query( "DROP TABLE IF EXISTS vergleich" );
$wpdb->query( "DELETE FROM datenbankname65
              WHERE post_type ='veranstaltung'" );
```

5.2.3 connection.php

Um eine Aktion an einer Datenbank durchzuführen, bedarf es einer Verbindung zu dieser. Zum Herstellen der Verbindung mit der benötigten Datenbank verwendet das Plugin die `connection.php` (

Anhang 4: `connection.php`). Die Datei wird in die `importdaten.php` (5.2.4) und die `tabellen.php` (5.2.7) eingebunden, wo SQL-Abfragen an die Datenbank benötigt werden. Eine neue

⁶³ Vgl. Lefebvre 2012, S.82

⁶⁴ Basiert auf vgl. Starr 2020

⁶⁵ Name der Datenbank im folgenden Code geändert in `datenbankname`

Verbindung wird über `new mysqli()` hergestellt. Dabei müssen Host, Nutzernamen, Passwort und die Datenbank angegeben werden. Falls keine Verbindung möglich ist, wird dem Nutzer „Verbindung fehlgeschlagen ausgegeben“.

5.2.4 importdaten.php

Die `importdaten.php` (Anhang 7: `importdaten.php`) ist die Kernfunktion des Plugins und sorgt dafür, dass die auf der Seite vorhandenen oder die selbst erstellten Veranstaltungen in die vom Plugin angelegte Datenbanktabelle per Klick importiert, gelöscht oder aktualisiert werden kann. Dabei wird die `importdata`-Funktion aufgerufen. Der Aufbau und die Funktionen der Datei sind in Abbildung 9 dargestellt.

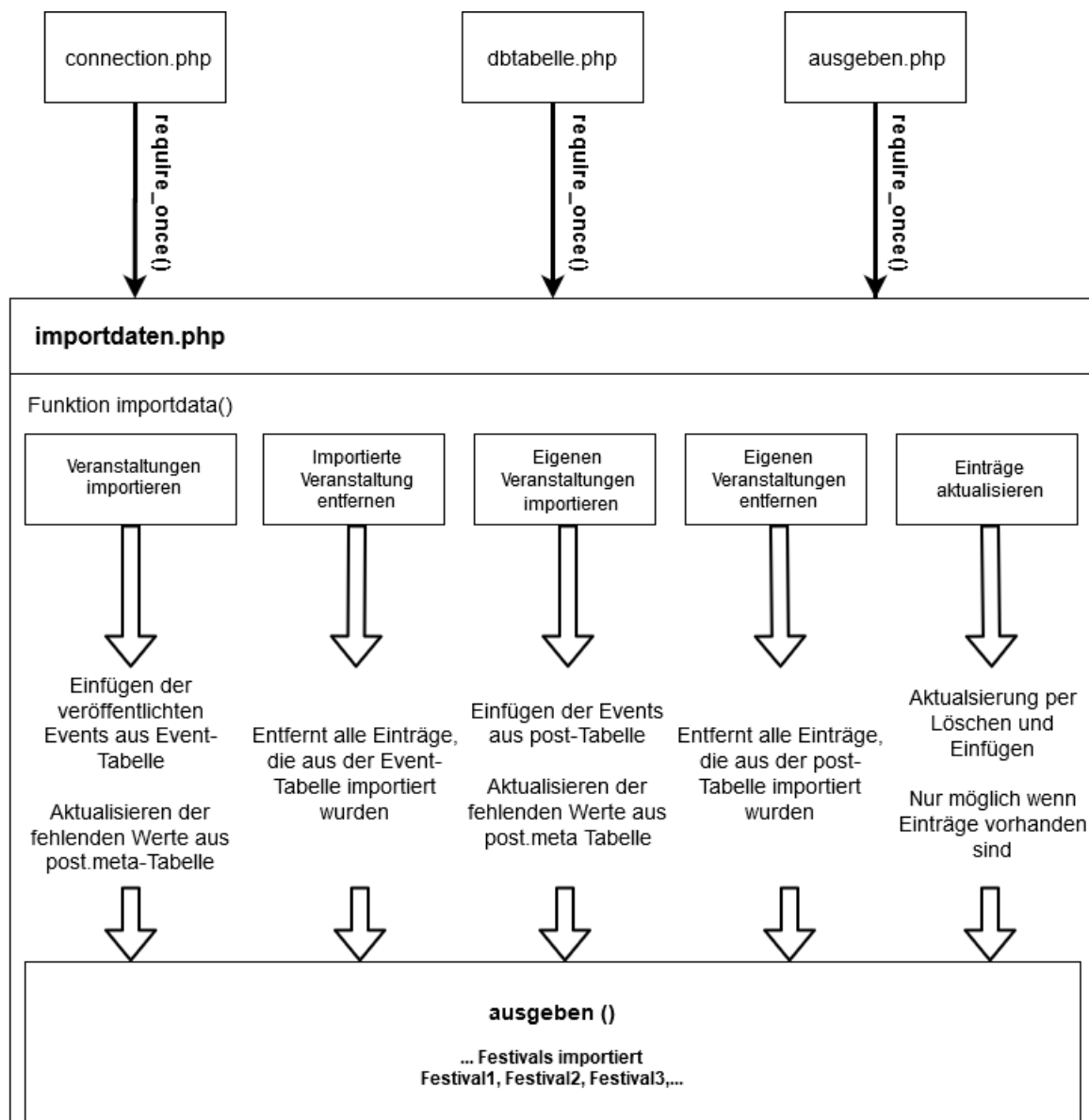


Abbildung 9: Aufbau Importfunktion⁶⁶

⁶⁶ Eigene Darstellung nach Anhang 7: `importdaten.php`

Durch das Einbinden der `dbtabelle.php` (5.2.5) wird geprüft, ob eine Datenbanktabelle für den Vergleich bereits vorhanden ist oder nicht. Dem Nutzer im Backend werden nun folgende Möglichkeiten geboten: Daten aus der Veranstaltungstabelle importieren, importierte Veranstaltungen löschen, eigene Veranstaltungen importieren, eigene Veranstaltungen löschen und Einträge aktualisieren. Das Importieren geschieht hauptsächlich über SQL-Abfragen. Die Festivals, die auf der Seite über das Plugin „Events Manager“ oder das CPT erstellt worden sind, befinden sich in der WordPress Datenbank an verschiedenen Orten. Während die CPT, wie Beiträge in der „posts“-Datenbanktabelle gespeichert sind, speichert das Plugin „Events Manager“ seine Veranstaltungen in einer eigenen Datenbanktabelle. Es werden allerdings nicht alle Inhalte in diesen Tabellen gespeichert. So sind Metadaten in der von WordPress angelegten Tabelle „postmeta“ hinterlegt. Veranstaltungen und Metadaten können mit einer ID klar einander zugeordnet werden.

Die Import-, Lösch- und Aktualisierungsfunktionen werden ausgeführt, wenn der entsprechende Button gedrückt wurde. Im Quellcode sind die Buttons in einem Formular angelegt, dass bei Aktivierung dieser Buttons, per Datenübergabe einen Wert setzt. Nach Setzen des Wertes werden dann gewünschte Befehle wie das Importieren oder Löschen ausgeführt.

Beim Import aus der Tabelle des Events Managers werden, zunächst per SQL-Befehl, die zu importierenden Datensätze aus der „Events Manager“-Datenbanktabelle⁶⁷ herausgefiltert. Hierbei ist es wichtig, dass nur veröffentlichte Veranstaltungen ausgewählt werden, da sonst in der späteren Vergleichstabelle auch Einträge von Veranstaltungen erscheinen, die im Backend im Papierkorb liegen oder als Entwurf gespeichert sind. Alle Datensätze, die den Anforderungen entsprechen, werden in die vom Plugin erstellte Datenbanktabelle eingefügt. Das zuvor beschriebene SQL-Statement ist im Quellcode 8 aufgeführt.

Quellcode 8: SQL-Befehl für den Import⁶⁸

```
(...)  
$sqlimport = "INSERT INTO vergleich (id, name, sdatum, edatum)  
              SELECT post_id, event_name, event_start_date, event_end_date  
              FROM datenbankname_em_events  
              LEFT JOIN datenbankname_posts  
              ON datenbankname_em_events.post_id= datenbankname_posts.ID  
              WHERE post_status = 'publish';  
(...)
```

⁶⁷ In folgenden als Eventtabelle beschrieben

⁶⁸ Vgl. Anhang 7: `importdaten.php`

Mit dem oben beschriebenen Vorgehen wird jedoch nur ein Teil der benötigten Daten importiert, die Informationen aus der “postmeta“-Tabelle fehlen noch. Durch einen Vergleich der ID wird, zu den zuvor eingefügten Festivals, der passenden Datensatz gefunden. Damit keine eigenen Festivals hinzugefügt werden, sondern nur die aus der Eventtabelle kommen genutzt werden, wird die Aktualisierung nur bei Elementen durchgeführt, die in der Importspalte der “vergleich“-Tabelle⁶⁹ nicht den Wert “veranstaltung“ stehen haben. Die entsprechende SQL-Abfrage zeigt Quellcode 9. Der Wert “veranstaltung“ wird eigenen Einträgen beim Import hinzugefügt. Das Aktualisieren geschieht für jede benötigte Spalte der späteren Vergleichstabelle, dessen Werte in der “postmeta“-Tabelle abgelegt sind. Die Ausführung der SQL-Abfragen geschieht, wenn der entsprechende Button gedrückt wird.

Quellcode 9: SQL-Befehl für Hinzufügen von Informationen beim Import⁷⁰

```
(...)  
$sqlbesucher = "UPDATE vergleich SET vergleich.besucher = (SELECT datenbankname_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id = datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'Zuschauer') WHERE `import` <>'veranstaltung'";  
(...)
```

Der Import von eigenen Veranstaltungen verläuft nahezu analog. Unterschiede sind im Quellcode 10 ersichtlich. Einzig die Selektion und Verbindungen der Datenbanktabellen sind ein wenig anders. CPT werden in der “posts“-Tabelle der WordPress-Datenbank gespeichert, wobei eine bei der Programmierung angegebene Bezeichnung in der Spalte “post_type“ mitgespeichert wird. Der Wert wird beim Importieren in die Spalte “import“ übernommen und sorgt dafür, dass die über das CPT erstellten Veranstaltungen, sich von den Festivals, die aus der Events-Tabelle importiert wurden, unterscheiden. Hinzukommt, dass nur Festivals eingefügt werden, die entweder veröffentlicht oder als Entwurf gespeichert worden sind. Im Gegensatz zum Import aus der Eventtabelle werden die als Entwurf gespeicherten Festivals bewusst mit ausgewählt, da bei jeder angelegten eigenen Veranstaltung zusätzlich eine Seite mit URL generiert wird. Sollen diese Seiten im Frontend nicht aufrufbar sein, empfiehlt es sich, den neu erstellten Eintrag auf Entwurf zu stellen.

⁶⁹ Vom Plugin erstellte Datenbanktabelle

⁷⁰ Vgl. Anhang 7: importdaten.php

Quellcode 10: SQL-Befehl für den Import der eigenen Festivals⁷¹

```
(...)  
$sqlleigen = "INSERT INTO vergleich (id, name, import) SELECT datenbank-  
name_posts.ID,post_title,post_type FROM datenbankname_posts WHERE post_type =  
'veranstaltung' AND (post_status = 'draft' OR post_status = 'publish')";  
(...)
```

Bei Ausführung der SQL-Abfragen via „Löschen“-Buttons, wird der Inhalt der „vergleich“-Tabelle gelöscht. Es wird dabei nach dem Eintrag der Datensätze in der Spalte „import“ unterschieden. Wird der Button für das Löschen der eigenen Beiträge gedrückt, werden nur Elemente ausgewählt und entfernt, die den Wert „veranstaltung“ in der Tabellenspalte „import“ stehen haben. Beim Entfernen der Elemente aus der Eventstabelle sind nur die Elemente ohne diesen Wert betroffen. Quellcode 11 zeigt die SQL-Abfragen zum Löschen der Elemente.

Quellcode 11: SQL-Befehl zum Löschen von Einträgen⁷²

```
(...)  
$sqldelete = "DELETE FROM `vergleich` WHERE `import` <>'veranstaltung';"  
$sqldeleteown = "DELETE FROM `vergleich` WHERE `import` = 'veranstaltung';"  
(...)
```

Nur wenn bereits Inhalte importiert wurden, ist das Aktualisieren möglich. Mit den SQL-Abfragen aus Quellcode 12 wird geprüft, wie viele Reihen in der Datenbanktabelle „vergleich“ vorhanden sind. Auch hier wird wieder zwischen den Veranstaltungen vom „Events Manager“ und den eigenen Veranstaltungen unterschieden. Das Aktualisieren erfolgt jedoch über einen gemeinsamen Button und nicht wie zuvor durch getrennte.

Quellcode 12: SQL-Befehl zum Zählen der Datensätze in der Tabelle⁷³

```
(...)  
$sqldatacheck=$db->query("SELECT COUNT(*) FROM vergleich WHERE `import`  
<>'veranstaltung'");  
$sqldatacheckown=$db->query("SELECT COUNT(*) FROM vergleich WHERE `im-  
port`='veranstaltung'");  
(...)
```

⁷¹ Vgl. Anhang 7: importdaten.php

⁷² Vgl. Anhang 7: importdaten.php

⁷³ Vgl. Anhang 7: importdaten.php

Die Werte, die bei der SQL-Abfrage herauskommen, werden in Arrays geschrieben, es wird ferner überprüft, ob sie größer als null sind. Ist das der Fall, werden die Einträge erneuert. Erneuern heißt in diesem Fall, das Leeren der Tabelle und erneutes Einfügen in die “vergleich“-Tabelle, wie beim Import. Im Quellcode 13 ist der Code für die Aktualisierung der Einträge zu sehen.

Quellcode 13: Aktualisierungsfunktion der Einträge⁷⁴

```
(...)  
else if(isset($_POST['update'])) {  
    while ($reihe = $sqldatacheck->fetch_row()) {  
        $countimport[] = array($reihe[0]);  
    }  
(...)  
    $anzahlimport = $countimport[0][0];  
(...)  
    if ($countimport[0][0]>0){  
        $db->query($sqldelete);  
        $db->query($sqlimport);  
        (...)  
    } else if ($anzahlimport==0){  
        echo "Keine Einträge aus Events-Datenbanktable  
vorhanden.<br>";  
    }  
(...)
```

Zum Abschluss der “importdata“-Funktion wird die “ausgeben“-Funktion aus der `ausgeben.php` (5.2.6) zwei Mal aufgerufen. Sie listet alle importierten Festivals auf. Diese werden getrennt in eigene Veranstaltungen und solche aus dem „Events Manager“-Plugin.

5.2.5 dbtabelle.php

Beim Öffnen der Importseite über das Menü wird die “importdata“-Funktion aufgerufen. Durch das Einbinden der `dbtabelle.php` (Anhang 6: `dbtabelle.php`) in die `importdaten.php` (5.2.4) wird der Inhalt der `dbtabelle.php` automatisch ausgeführt. Sollte bereits eine Datenbank vorhanden sein, bekommt der Nutzer die Ausgabe „Plugin ist für den Import bereit“ ausgegeben. Ist keine Datenbank mit dem Namen “vergleich“ vorhanden, wird eine Tabelle

⁷⁴ Vgl. Anhang 7: `importdaten.php`

per SQL-Abfrage erstellt. Dabei bekommt die Tabelle ihre Spalten mit den jeweiligen Eigenschaften. Neben den Veranstaltungsdaten wie Name, Datum, Preis Camping, Bands, Besucher und ID, wird außerdem die Spalte "Import" der Datenbanktabelle hinzugefügt. Diese dient später dazu, die aus der Veranstaltungsdatenbank importierten Datensätze von den selbst erstellten zu unterscheiden. Das Attribut Name ist als Primärschlüssel der Tabelle deklariert. Ein Primärschlüssel wird als eindeutiger Identifikation eines Datensatzes benötigt und kann aus einen oder mehreren Attributen bestehen.⁷⁵ Im Falle von Festivals reicht der Name zur eindeutigen Identifizierung aus. Die Abbildung 10 stellt die Funktion, der Datei da. Der Primärschlüssel ist unterstrichen.

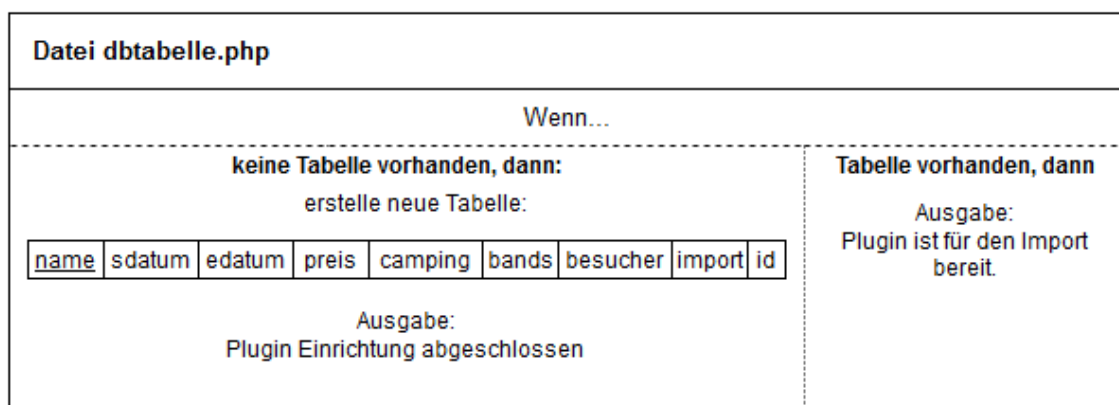


Abbildung 10: Datenbanktabelle erstellen⁷⁶

5.2.6 ausgeben.php

Die ausgeben.php (Anhang 3: ausgeben.php) wird von der importdaten.php (5.2.4) eingebunden und zur Ausgabe des Inhalts der Vergleichsdatenbank im Menüpunkt Import verwendet. Eine Übersicht, was während des Ausführens der Funktion passiert ist in Abbildung 11 verbildlicht. Wird die Funktion "ausgeben" aufgerufen, muss eine Variable übergeben werden. Bei dieser Variablen handelt es sich um ein zuvor in der importdaten.php definiertes SQL-Abfrageergebnis. Die Funktion überprüft zunächst ob der Inhalt der Variable gleich oder ungleich null ist. Ist die Variable gleich null, wird über die nicht vorhandenen Daten informiert. Befinden sich Datensätze in der Variablen, wird eine Variable mit dem Namen "zaehler" angelegt, die später zur Zählung der Elemente dient. Zudem werden mittels einer *while-Schleife*, eine Kontrollstruktur, die solange eine Anweisung ausführt, bis eine Bedingung erfüllt ist, Datensätze aus der übergebenen Variablen in ein mehrdimensionales Array geschrieben. Nach jedem Datensatz wird die "zaehler"-Variable um eins erhöht. Dadurch wird die Gesamtanzahl der Elemente in dem Array ermittelt. Als sichtbares Ergebnis werden

⁷⁵ Vgl. Schicker 2017, S.31

⁷⁶ Eigene Darstellung nach Anhang 6: dbtabelle.php

die Anzahl sowie eine Auflistung der importierten Festivals durch eine *for-Schleife*, eine Wiederholung von Anweisungen bis eine bestimmte Bedingung erreicht wurde, ausgegeben.

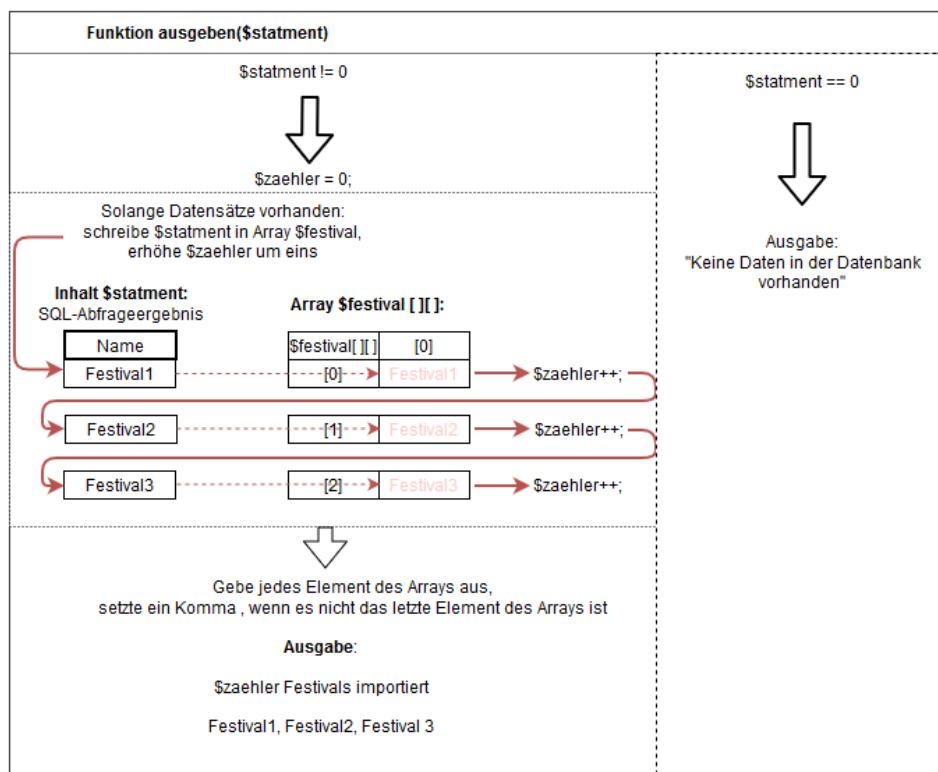


Abbildung 11: Funktion Tabelle ausgeben⁷⁷

5.2.7 tabelle.php

Die tabelle.php (Anhang 9: tabelle.php) ist dafür zuständig, dass nach dem Eintragen des Shortcodes auf einer Seite oder in einem Beitrag die Vergleichstabelle erscheint. Nutzer können so verschiedenen Festivals auswählen und diese per Klick miteinander vergleichen.

Für die korrekte Darstellung der Informationen werden neben der Verbindung zur Datenbank noch die Funktionen tage.php (5.2.10) und datumformat.php (5.2.9) eingebunden. Ähnlich wie in der ausgeben.php (5.2.6) werden die Datensätze aus dem SQL-Abfrageergebnis mit allen Einträgen der Tabelle "vergleich", in alphabetischer Reihenfolge, in einem mehrdimensionalen Array gespeichert. Ebenfalls wird eine "zaehler"-Variabel angelegt, die mit jedem neuen Element im Array hochgezählt wird, um später die Anzahl der Elemente zu erhalten. Durch eine *while-Schleife()* wird jeder Datensatz der Datenbank in eine Reihe des mehrdimensionalen Arrays geschrieben. An der Stelle "\$reihe[3]" wird dem Element ein Eurozeichen angehängen, da es sich hier um den Preis des Festivals handelt. Dieses

⁷⁷ Eigene Darstellung nach Anhang 3: ausgeben.php

geschieht solange, bis alle Datensätze des SQL-Abfrageergebnisses durchlaufen sind. Die Umsetzung dieses Vorgangs ist in Quellcode 14 abgebildet.

*Quellcode 14: Datenbanktabelle wird in ein Array geschrieben*⁷⁸

```
(...)  
$sql = "Select * from vergleich ORDER BY `name` ASC";  
$zaehler = 0;  
if ($r = $db->query($sql)) {  
    while ($reihe = $r->fetch_row()) {  
        $festival[] = array($reihe[0], $reihe[1], $reihe[2], $reihe[3]."  
&euro;", $reihe[4], $reihe[5], $reihe[6], $reihe[7]);  
        $zaehler++;  
    }  
}  
(...)
```

Um für den Nutzer eine Tabelle zu generieren, in der alle importierten Festivals aufgelistet sind, wird eine Tabelle und ein Formular verwendet. In der Kopfzeile der Tabelle stehen die Beschreibung der angezeigten Informationen: Festivalname, Start- und Enddatum, die Anzahl der Tage, der Ticketpreis, ob Camping möglich ist, wie viele Bands auftreten und wie viele Besucher das Festival bei der letzten Durchführung hatte. Unter der Kopfzeile können bis zu acht Festivals ausgewählt und verglichen werden. In der ersten Spalte der Tabelle gibt es ein Auswahlfeld, in das per *for-Schleife()* Optionen hinzugefügt werden, bis die Variable "i" innerhalb der Schleife nicht mehr kleiner als die Anzahl der Festivals im Array ist. Diese Werte der Optionen werden aus dem Array an der Stelle [*\$i*][0] ausgelesen. Das heißt, dass das Auswahlfeld den Namen jedes Festivals im Array als Auswahloption angibt. Der Quellcode 15 zeigt die Generierung der Auswahloptionen für die Vergleichstabelle. Bei den Spalten zwei und drei der Tabelle wird durch die Funktion "format", das übergebene Datum in die in Deutschland gängige Schreibweise umgewandelt. In der vierten Spalte übernimmt die Funktion "tage" das Ausrechnen der Festivaltage mittels Start- und Enddatums.

*Quellcode 15: Auswahloptionen für die Tabelle*⁷⁹

```
(...)  
for($i= 0; $i < $zaehler; $i++){  
    ?><option value="<?php echo $i ?>"><?php echo $festival[$i][0] ?></option>
```

⁷⁸ Vgl. Anhang 9: tabelle.php

⁷⁹ Vgl. Anhang 9: tabelle.php

(...)

Ist vom Nutzer eine Auswahl getätigt und der Formularbutton unter der Tabelle gedrückt worden, werden ausgewählte Werte mit `$_POST` im Formular per Datenübergabe übergeben und erscheinen in der Tabelle. Beim Erstellen der Auswahloptionen wird die Variable `“i“` als `“value“` für die jeweilige Option gespeichert. Wählt der Nutzer beispielsweise im vierten Auswahlfeld, das als `“festival4“` deklariert ist, eine Veranstaltung aus und klickt auf den Button, wird der gespeicherte Wert in den anderen Spalten, in denen im Quellcode der Name des Auswahlfeldes auftaucht, eingesetzt. Die Ausgabe des Startdatum und des Preises werden beispielsweise im Quellcode 16 gezeigt. Dabei werden Spalte für Spalte die zum Festival gehörigen Informationen ausgegeben.

Quellcode 16: Ausgabe der Tabelle⁸⁰

```
(...)  
<td data-title="Start">  
    <?php echo format($festival[$_POST['festival4']][1]);?>  
</td>  
(...)  
<td data-title="Preis">  
    <?php echo $festival[$_POST['festival4']][3];?>  
</td>  
(...)
```

Abbildung 12 zeigt die fertige Vergleichstabelle, die im Frontend durch die `“erzeugetabelle“`-Funktion erstellt und aufgerufen wird.

Festivalname	Start	Ende	Tage	Preis	Camping	Bands	Besucher
Deichbrand Festival	16.07.20	19.07.20	4	139 €	Ja	70	60000
Heimatzoo Festival	07.08.20	08.08.20	2	18 €	Ja	0	10
Highfield Festival	14.08.20	16.08.20	3	119 €	Ja	40	35.000
Hurricane Festival	19.06.20	21.06.20	3	149 €	Ja	70	70.000
Parookaville	17.07.20	19.07.20	3	209 €	Ja	300	210.000
Rock am Ring	05.06.20	07.06.20	3	149 €	Ja	75	85.000
Reeperbahn Festival	16.09.20	19.09.20	4	99 €	Nein	600	50.000
Southside Festival	19.06.20	21.06.20	3	159 €	Ja	70	60.000

Festivals auswählen

Abbildung 12: Vergleichstabelle im Frontend⁸¹

⁸⁰ Vgl. Anhang 9: tabelle.php

⁸¹ Vgl. Konzertkalender.net 2020c

5.2.8 tabelle.css

Das tabelle.css (Anhang 11: tabelle.css) gibt der Tabelle im Frontend ihr Aussehen und ihre Eigenschaften. Auf Bildschirmen mit großer Auflösung, ist das Darstellen der Tabelle kein Problem. Schwierig wird die Darstellung auf kleineren Geräten, da die Tabelle durch ihre Breite, auf deutlich höher als breiteren Geräten nicht in Gänze dargestellt werden kann. Hierbei geht im Fall dieses Plugins die Übersichtlichkeit verloren. Normalerweise sollten zu vergleichende Eigenschaften untereinanderstehen, damit diese schnell abgeglichen werden können. Allerdings würde ohne responsive Anpassung die Tabelle einfach über den vorgesehenen Bereich hinausragen. Eine optimale Nutzung wäre so nicht möglich.

Für die responsive Optimierung werden den Tabellenelementen die Eigenschaft “display: block“ gegeben. Die Tabellenzeilen werden nicht mehr nebeneinander, sondern untereinander abgebildet. Hinzu kommt, dass die Inhalte des Tabellenkopfes ausgeblendet werden und beim Inhalt der Tabelle wiederauftauchen sollen. Mehrere “data“-Attribut in der HTML-Tabelle, in denen der jeweilige Inhalt der Spalten des Tabellenkopfes angegeben ist, werden ausgelesen und mit der “content“-Eigenschaft wiedergegeben.⁸²

Quellcode 17: tabelle.css⁸³

```
(...)  
@media only screen and (max-width: 760px), (min-device-width: 768px) and (max-  
device-width: 1024px){  
    table, thead, tbody, th, td, tr {  
        display: block;  
        text-align: left;  
    }  
    thead {  
        display: none;  
    }  
(...)  
    td:before {  
        content: attr(data-title);  
        display: inline-block;  
        width: 35%;  
        padding-right: 10px;  
        white-space: nowrap;  
    }  
}
```

⁸² Vgl. Grochtdreis 2014

⁸³ Basiert auf vgl. Grochtdreis o.J.

(...)

Neben den Anpassungen von Tabelleneigenschaften wird durch die Deklaration von Media Querys, durch @media die Darstellung an unterschiedliche Ausgabegeräte angepasst. Eine Differenzierung wird durch weitere Eigenschaften, wie die Gerätebreite, Auflösung und Ausrichtung und die Angabe von minimalen oder maximalen Werten erwirkt.⁸⁴ In tabellen.css sind diese Werte für mehrere Ausgabegeräte vordefiniert. Einen Ausschnitt, wie die CSS-Datei die Tabelle an andere Bildschirmgrößen anpasst, zeigt Quellcode 17.

Die Einbindung der CSS-Datei erfolgt in der index.php (5.2.1) über einen Action-Hook.

Wird die Vergleichstabelle beispielsweise über ein Handy aufgerufen, passt sich die Tabelle, sichtbar in Abbildung 13, an das Gerät an.

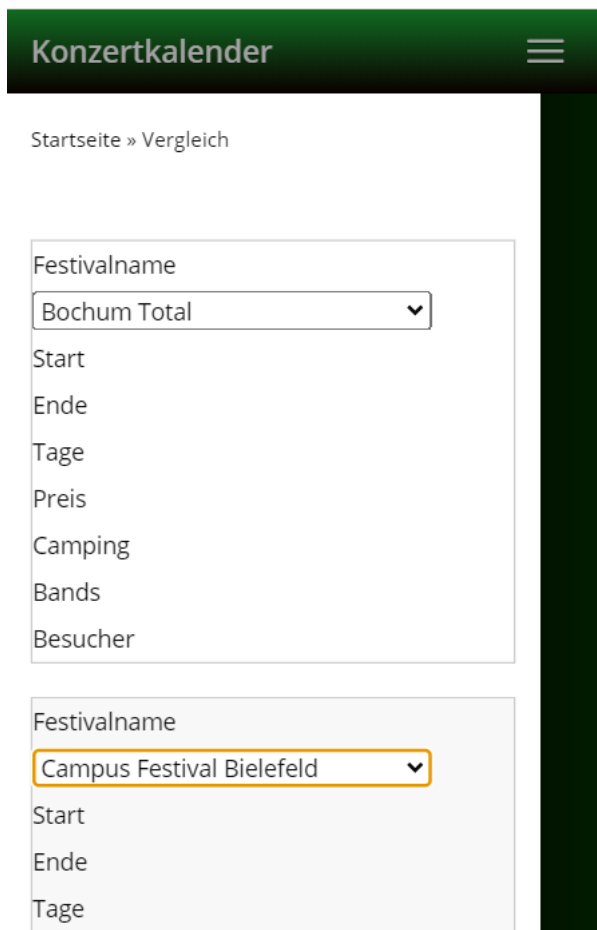


Abbildung 13: Vergleichstabelle (Handy)⁸⁵

⁸⁴ Vgl. Bühler u. a. 2017, S.76-77

⁸⁵ Vgl. Konzertkalender.net 2020c

5.2.9 datumformat.php

Alle Daten der Veranstaltung werden in der Datenbank im Format Jahr-Monat-Tag gespeichert. In diesem Format sind sie allerdings für die Darstellung in der Tabelle nicht brauchbar und müssen in das in Deutschland gängige Format Tag-Monat-Jahr umgewandelt werden. Diese Umwandlung kann mittels der Funktion “format“ der datumformat.php (Anhang 5: datumformat.php) erfolgen. Die Funktion benötigt ein Datum zu Verarbeitung, das in drei neue Variablen: Tag, Monat und Jahr aufgespalten wird. Über die String-Funktion *substr()* werden nur bestimmte Bestandteile des Datums in den jeweiligen Variablen gespeichert. So werden in der Tag-Variable beispielsweise nur die beiden letzten Zeichen des Datums gespeichert. Die einzelnen Bausteine werden dann in einer neuen Variable “date“ zum Datum im gewünschten Format zusammengesetzt und bei Aufruf ausgegeben. In Abbildung 14 wird bildlich die Funktionsweise der “format“-Funktion dargestellt.

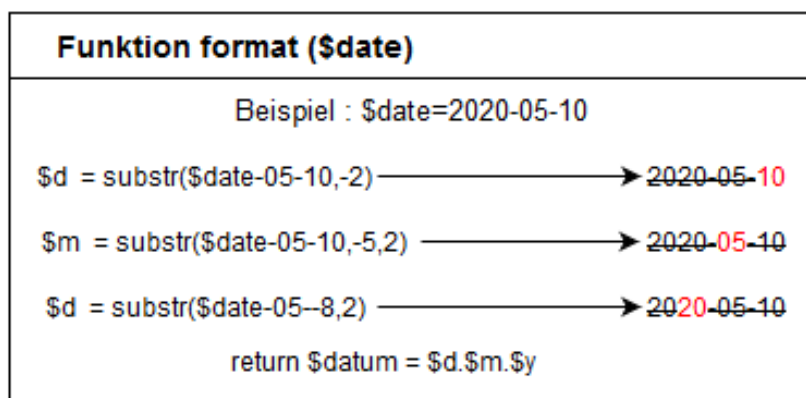


Abbildung 14: Funktion Datum umwandeln⁸⁶

5.2.10 tage.php

In der fertigen Tabelle wird die Anzahl der Tage des Festivals abgebildet, obwohl diese nicht in der Datenbanktabelle gespeichert sind. Die Berechnung der Dauer der Veranstaltung in Tage erfolgt über die “tage“-Funktion der tage.php (Anhang 10: tage.php). Bei der Ausgabe in der Frontend-Tabelle wird die Funktion aufgerufen und nimmt das Start- und Enddatum des Events entgegen. Um auf die exakte Anzahl der Tage zu kommen, wird das Enddatum durch die *strtotime()* in einen UNIX-Zeitstempel in die Einheit Sekunden umgewandelt und um einen Tag in Sekunden erhöht. Darauf folgt eine erneute Umwandlung in das vorherige Datumsformat durch *date()*. Das Startdatum und das neue Enddatum werden nun als eine neue *DateTime()* gespeichert. Schließlich wird in einer neuen Variablen die Differenz

⁸⁶ Eigene Darstellung nach Anhang 5: datumformat.php

zwischen den beiden Daten gespeichert und in ein passendes Ausgabeformat gewandelt. Ein Beispiel zur Rechnung innerhalb der Funktion ist auf Abbildung 15 zu sehen.

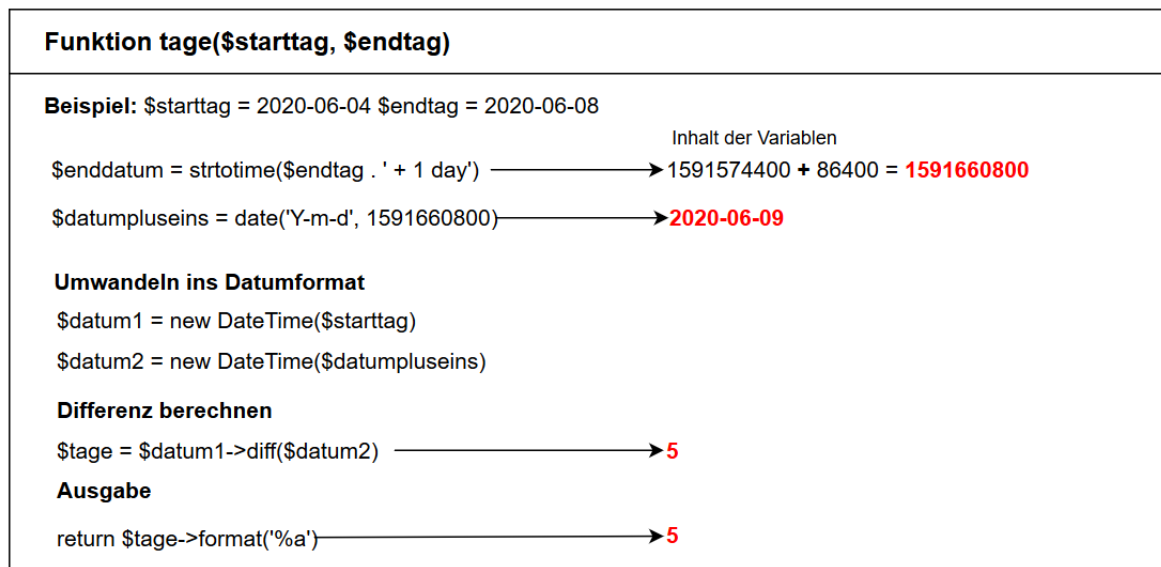


Abbildung 15: Tage errechnen⁸⁷

5.2.11 startseite.php

Die Startseite erscheint, wenn im Menü auf den Menüpunkt des Plugins geklickt wird. In Abbildung 16 ist die Anleitung zu sehen. Hier wird detailliert beschrieben, wie das Plugin funktioniert und welche Möglichkeiten es bereitstellt. Das Aufrufen der “anleitung“-Funktion aus der startseite.php (Anhang 8: startseite.php) geschieht dabei über die index.php (5.2.1).

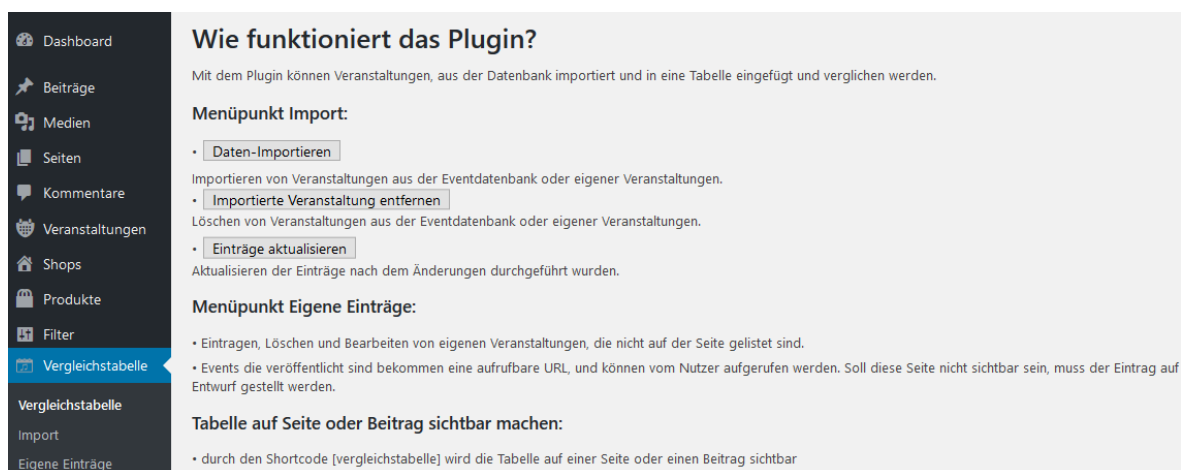


Abbildung 16: Anleitung des Plugins im Backend⁸⁸

⁸⁷ Eigene Darstellung nach Anhang 10: tage.php

⁸⁸ Backend konzertkalender.net (Bild durch Autor bearbeitet)

6 Auswertung

6.1 Entwurf und Realisierung

Im Vergleich zum eingangs geplanten Entwurfs gab es bei der finalen Realisierung ein paar Abweichungen. Bezüglich des Aufbaues und der Darstellung der Tabelle hat sich nichts geändert. Es können hier bis zu acht Festivals anhand von sieben Parametern verglichen werden. Im Backend hat das Plugin wie geplant einen eigenen Menüpunkt. Während der Umsetzung, sind aus den geplanten vier Untermenüpunkten jedoch zwei geworden. Der Menüpunkt Löschen entfällt, da die vom Plugin erstellten Dateien durch die `uninstall.php` (5.2.2) beim Entfernen des Plugins automatisch gelöscht werden. Die Seite für den Shortcode wurde in die Anleitung umgelagert und ist sichtbar, wenn auf den Hauptmenüpunkt geklickt wird. Bei den realisierten Menüpunkten ist unter Import das Aktualisieren der Einträge hinzugekommen, welches die Möglichkeit bietet einzelne Einträge zu löschen oder zu entsetzen. Wird eine Veranstaltung entweder über das Plugin „Events Manager“ oder über den Plugin eigenen CPT gelöscht, erscheint diese nach einem Klick auf „Aktualisieren“ nicht mehr als Eintrag in der Tabelle. Beim Menüpunkt Einfügen ersetzt der angelegte CPT das zuvor eingelebte geplante HTML-Formular. Da das CPT das Formular in der Metabox integriert hat.

6.2 Testlauf

Der Testlauf erfolgt über eine exakte Kopie der Seite `konzertkalender.net`. Im Backend von WordPress wird das Plugin über den Menüpunkt Plugin/Installieren auf die Seite geladen. Während des Installierens und Aktivierens des Plugins treten keine Fehler auf. Das Veranstaltungsplugin erscheint nun, wie in Abbildung 17 zu sehen, in der Liste der installierten Plugins.



Abbildung 17: Eintrag in Liste der installierten Plugins⁸⁹

⁸⁹ Backend `konzertkalender.net`

Um die Import-Funktion auch an den eigenen Veranstaltungen zu testen, wird im Untermenüpunkt „Eigene Einträge“ des Plugins, ein neuer Eintrag erstellt.

The screenshot shows a form for creating a festival entry. The title is 'Test-Festival'. Below the title is a permalink: https://staging.konzertkalender.net/plugin_test/veranstaltung/test-festival/ with a 'Bearbeiten' button. The form is divided into two main sections: 'Informationen' and 'Veröffentlichen'.

Informationen:

- Von: 25.06.2020
- Bis: 30.06.2020
- Preis: 24
- Anzahl Bands: 15
- Besucher: 2500
- Camping: Ja

Veröffentlichen:

- Status: Entwurf [Bearbeiten](#)
- Sichtbarkeit: Öffentlich [Bearbeiten](#)
- Sofort veröffentlichen [Bearbeiten](#)
- Lesbarkeit: Nicht verfügbar
- SEO: Nicht verfügbar
- Buttons: Cache leeren, [In den Papierkorb verschieben](#), [Veröffentlichen](#)

Abbildung 18: Eigene Eintrag erstellen⁹⁰

Nach Durchführung der Importfunktion, erscheint eine Liste mit der Anzahl der eigenen und der durch den „Events Manager“ angelegten und importierten Festivals. Abbildung 19 zeigt diese Liste. Der zuvor erstellte Eintrag in Abbildung 18 des „Test-Festivals“ wurde ebenfalls erfolgreich importiert.

The screenshot shows two sections of a festival list:

Liste der eigenen Festivals:

- 1 Festivals importiert
- Test-Festival

Liste der importierten Festivals aus Events-Datenbanktabelle:

- 118 Festivals importiert
- A Summer's Tale, About You Pangea Festival, Acoustic Winter Festival, Aftersight & Buten, Airbeat One Festival, Altheimer Open-Air, Andy ist auf einem Tennisturnier, Angletree Garden, Artlake Festival, Autumn Moon Festival, Baltic Open Air, BigCityBeats World Club Dome, BigCityBeats World Club Dome Winter Edition, Bochum Total, Brennale, Burg Herzberg Festival, c/o pop Festival, Campus Festival Bielefeld, Campus in Concert, Campusfestival Mittweida, Cologne Musik Week, DamsRock 2020, Das Fest Karlsruhe, Deichbrand Festival, Die Festung Rockt, Dreyland-Bluesfestival, E-Tropolis, Echelon Open Air, Eier mit Speck Festival, Elbjazz, Electrinity Festival, Electrique Baroque, Fahrmanntest, Feel Festival, Full Force Festival, Fusion Festival, Getöse in Moese Festival, Golden Leaves Festival, Green Juice Festival, Gut Sandbeck Open Air, Hafen Rock, Haldern Pop Festival, Happiness Festival, Haselnüsse Bühnen - Singer Songwriter Festival, Heimatzo Festival, Helene Beach Festival, Highfield Festival, Hunteburger Open Air Festival, Hurricane Festival, Hütte Rockt Festival, ImmerGrün Musikfestival, Immergut Festival, Jamel rockt den Förster, Juicy Beats Festival, Kapelle Somma 2020, Kleinstadtfestival, Let The Bad Times Roll Open Air, Lollapalooza Berlin, MAYDAY Dortmund, MELTI Festival, Metal Hammer Paradise, Mosaïque Festival, Moyn Moyn Festival, MS Dockville Festival, M'era Luna Festival, N-Joy Starshow, Nature One, NDR2 - Plaza Festival, New Fall Festival, Oldenbora, Oldenburger Kultursommer, Open Flair Festival, Open R Festival Neue Töne, Parookaville, pink November Musikfestival, Plage Noire, Pop-Kultur Berlin, Rieseperbahn Festival, Reggae Jam Festival, Reload Festival, Rocco del Schlacko, Rock am Beckenrand, Rock am Deister, Rock am Ring, Rock den Deich, Rock im Park, Rocken am Brocken, Rolling Stone Beach, Rolling Stone Park, Rudolstadt-Festival, Ruhr-in-Love, Ruhrpott Rodeo, Sacred Ground Festival, Sea You Festival, Sonne, Mond und Sterne Festival, Sound of the Forest Festival, Southside Festival, Spektrum Festival, SPLASH! Festival, Stadt ohne Meer-Festival, Stockrock Festival, Südwinsen Festival, Summerjam, Summerme Festival, SWR3 New Pop Festival, TabulaRazza Festival, Talge Open Air, Tante Mia Tanz, Taubertal Festival, Vainstream Rockfest, Venner Folkfrühling, Volle Kraft Voraus Festival, Wacken Open Air, Wacken Winter Nights, Watt En Schlick Festival, Wave-Gotik-Treffen, Wilde Möhre Festival, Ziegelei Open Air

Abbildung 19: Liste der importierten Festivals⁹¹

Im Frontend sollte die Tabelle mit den aufgelisteten Festivals erscheinen. Voraussetzung dafür ist, dass der Shortcode per Editor auf einer Seite mit dem Namen Vergleich eingefügt wurde. Beim Speichern der Seite tritt jedoch ein Problem auf. Anstelle wie üblich auf der Bearbeitungsseite des Beitrages zu landen, erscheint ein weißer Bildschirm mit dem unformatierten Inhalt der Seite. Der Fehler liegt vermutlich in der tabelle.php (Anhang 9: tabelle.php). Eine Fehlerrecherche ergibt, dass Funktionen zum Senden oder Ändern von HTTP-Headern vor jeder Ergebnis-Ausgabe aufgerufen werden müssen. Eine Lösung des Fehlers im Backend ist bis dato noch nicht gefunden worden. An der Funktionalität der Plugins ändert der Fehler nichts und der Beitrag wird trotzdem gespeichert. Ruft man die

⁹⁰ Backend konzertkalender.net

⁹¹ Backend konzertkalender.net

Seite mit der Vergleichstabelle auf, wird diese, wie in Abbildung 20, korrekt und mit dem importierten Inhalt dargestellt.

Festivalname	Start	Ende	Tage	Preis	Camping	Bands	Besucher
Test-Festival	25.06.20	30.06.20	6	24 €	Ja	15	2500
Rock am Ring	05.06.20	07.06.20	3	149 €	Ja	75	85.000
Rock im Park	05.06.20	07.06.20	3	199 €	Ja	75	72.500
Deichbrand Festival	16.07.20	19.07.20	4	139 €	Ja	70	60000
Wacken Open Air	30.07.20	01.08.20	3	239 €	Ja	218	85.000
Hurricane Festival	19.06.20	21.06.20	3	149 €	Ja	70	70.000
Rock am Deister	25.07.20	25.07.20	1	10 €	Nein	6	800
NDR2 - Plaza Festival	29.05.20	29.05.20	1	41 €	Nein	4	25.000

Festivals auswählen

Abbildung 20: Vergleichstabelle im Frontend⁹²

Für den Test der Aktualisierungsfunktion wird „Rock am Ring“, eines der durch den „Events Manager“ erstellten Festivals, gelöscht. Zudem wird ein neuer eigener Eintrag mit dem Namen „Hannover-Festival“ hinzugefügt und die zuvor eingetragene Preis des „Test-Festivals“ geändert. Nach dem Aktualisieren erscheint in der Liste der eigenen Festivals das neu erstellte Festival. Der Eintrag Rock am Ring ist jetzt nicht mehr gelistet. In Rot markiert sind auf Abbildung 21 nun die aktualisierten Festivals.

Liste der eigenen Festivals:

2 Festivals importiert

Hannover-Festival, Test-Festival

Liste der importierten Festivals aus Events-Datenbanktabelle:

117 Festivals importiert

A Summer's Tale, About You Pangea Festival, Acoustic Winter Festival, Afdreht & Buten, Airbeat One Festival, Alzheimer Open-Air, Andy ist auf einem Tennisturnier, Appletree Garden, Artlake Festival, Autum Moon Festival, Baltic Open Air, BigCityBeats World Club Dome, BigCityBeats World Club Dome Winter Edition, Bochum Total, Breminale, Burg Herzberg Festival, c/o pop Festival, Campus Festival Bielefeld, Campus in Concert, Campusfestival Mittweida, Cologne Musik Week, DamsRock 2020, Das Fest Karlsruhe, Deichbrand Festival, Die Festung Rockt, Dreyland-Bluesfestival, E-Tropolis, Echelon Open Air, Eier mit Speck Festival, Elbjazz, Electrify Festival, Electricque Baroque, Fahrmannsfest, Feel Festival, Full Force Festival, Fusion Festival, Getoese in Moese Festival, Golden Leaves Festival, Green Juice Festival, Gut Sandbeck Open Air, Hafen Rock, Halderm Pop Festival, Happiness Festival, Haselünner Bühnen - Singer Songwriter Festival, Heimatzoo Festival, Helene Beach Festival, Highfield Festival, Hunteburger Open Air Festival, Hurricane Festival, Hütte Rockt Festival, ImmerGrün Musikfestival, Immergut Festival, Jamel rockt den Förster, Juicy Beats Festival, Kapelle Somma 2020, Kleinstadtfestival, Let The Bad Times Roll Open Air, Lollapalooza Berlin, MAYDAY Dortmund, MELT! Festival, Metal Hammer Paradise, Mosaïque Festival, Moyn Moyn Festival, MS Dockville Festival, M'era Luna Festival, N-Joy Starshow, Nature One, NDR2 - Plaza Festival, New Fall Festival, Oldenbora, Oldenburger Kultursommer, Open Flair Festival, Open R Festival, Neue Töne, Parookaville, pink November Musikfestival, Plage Noire, Pop-Kultur Berlin, Reeperbahn Festival, Reggae Jam Festival, Reload Festival, Rocco del Schlacko, Rock am Beckenrand, Rock am Deister, Rock den Deich, Rock im Park, Rocken am Brocken, Rolling Stone Beach, Rolling Stone Park, Rudolstadt-Festival, Ruhr-in-Love, Ruhrpott Rodeo, Sacred Ground Festival, Sea You Festival, Sonne, Mond und Sterne Festival, Sound of the Forest Festival, Southside Festival, Spektrum Festival, SPLASH! Festival, Stadt ohne Meer-Festival, Stockrock Festival, Südwinen Festival, Summerjam, Summertime Festival, SWR3 New Pop Festival, TabulaRaaza Festival, Talge Open Air, Tante Mia Tanz, Taubertal Festival, Vainstream Rockfest, Venner Folkfrühling, Volle Kraft Voraus Festival, Wacken Open Air, Wacken Winter Nights, Watt En Schlick Festival, Wave-Gotik-Treffen, Wilde Möhre Festival, Ziegelei Open Air

Abbildung 21: Liste der importieren Festivals⁹³

Beim Blick in das Frontend erscheint in der Auswahlliste das neu angelegte Festival. Das gelöschte Festival ist nicht mehr eingetragen. Die Aktualisierung des Preises beim Test-

⁹² Vgl. Konzertkalender.net 2020c

⁹³ Backend konzertkalender.net (Bild durch Autor bearbeitet)

Festival war ebenfalls erfolgreich. Durch einen Blick auf Abbildung 22 ist erkennbar, dass der Preis aktualisiert wurde.

Festivalname	Start	Ende	Tage	Preis	Camping	Bands	Besucher
Test-Festival	25.06.20	30.06.20	6	10 €	Ja	15	2500
Hannover-Festival	26.06.20	27.06.20	2	24 €	Nein	5	10000
			0				

Abbildung 22: Vergleichstabelle im Frontend⁹⁴

Bis auf den Fehler beim Speichern der Seite, der die Funktionalität des Ganzen aber nicht beeinflusst, gab es beim Testlauf keine weiteren Probleme oder Auffälligkeiten. Alle Funktionen sind ausführbar und erfüllen ihre Aufgaben. Die Hauptfunktion, das Importieren, geschieht schnell und ohne Probleme.

6.3 Veröffentlichung

Der Festivalvergleich ist online abruf- und nutzbar⁹⁵. Das Plugin wird vorerst nicht zum freien Download bzw. für andere WordPress-Nutzer zur Verfügung stehen. Dafür ist es im Moment noch zu speziell auf die Seite konzertkalender.net angepasst ist und würde in der jetzigen Form nicht auf anderen WordPress-Seite funktionieren. Zudem muss der bestehende Fehler vorher dringend behoben werden.

⁹⁴ Vgl. Konzertkalender.net 2020c (Bild durch Autor bearbeitet)

⁹⁵ Aufrufbar unter <https://www.konzertkalender.net/vergleich/>

Fazit und Ausblick

Das Ziel der Arbeit war die Konzeption eines lauffähigen WordPress-Plugins, das Nutzern auf übersichtliche Art und Weise in Form einer Tabelle hilft, Veranstaltungen zu vergleichen. Dabei sollte der Kern des Plugins eine Importfunktion sein, die bereits bestehende Veranstaltungen in diese Tabelle einfügen kann. Nach der Realisierung ist zu klären, ob durch die Umsetzung eine hilfreiche Erweiterung geschaffen wurde oder ob das Eintragen per Hand die bessere Methodik ist.

Final entstand im Zuge dieser Arbeit ein in die Internetseite konzertkalender.net integriertes, voll funktionsfähiges und nutzbares Plugin für den Vergleich von Veranstaltungen. Neben der Tabelle, die durch das Einfügen eines Shortcode auf einer Seite oder in einem Beitrag, aufgerufen wird und im Frontend erscheint, sind im Backend von WordPress die zugehörigen Funktionen aufrufbar. Die Hauptfunktion ist der Import von bereits vorhandenen Veranstaltungen in die Tabelle. Darüber hinaus erstellt das Plugin einen CPT mit dem es möglich ist, eigene Einträge zu erstellen und dann ebenfalls mit der Importfunktion in die Tabelle einzufügen. Hinzu kommt, dass das Plugin die Formatierung der Tabelle auch auf mobilen Endgeräten übernimmt. Außerdem können aus der Liste der importierten Veranstaltungen bis zu acht ausgewählt und verglichen werden.

Ohne das Plugin würde das Eintragen der Veranstaltungen in eine Tabelle per Hand geschehen, welches bei den über 100 eingetragenen Festivals auf der Seite konzertkalender.net einen nicht unwesentlichen Zeitaufwand nach sich ziehen würde. Auch mit dem Plugin und der „Eigene Einträge“-Funktion würde der Vorgang nicht schneller ablaufen. Durch das Ausführen des Importes werden alle vorhandenen Veranstaltungen in wenigen Sekunden in die Vergleichstabelle übertragen, ohne eine einzige Eintragung über das Vergleichsplugin tätigen zu müssen. Der Testdurchlauf zeigt, dass der Import deutlich schneller und mit weniger Arbeit verbunden ist. Im Großen und Ganzen ist das Plugin eine sinnvolle Erweiterung und stellt im Vergleich zum händischen Eintragen einer solchen Anzahl von Veranstaltungen die praktischere und komfortablere Methode da. Selbst bei einer deutlich geringeren Anzahl an Veranstaltungen kann das manuelle Eintragen zeitlich nicht mit dem Import mithalten.

Das Plugin umgeht zwar die redundante Dateneingabe, jedoch wird durch die momentane Umsetzung, die benötigten Eintragungen aus mehreren Datenbanktabelle kopiert und in einer andere einzelnen eingefügt. Es entstehen dabei doppelte Daten. Auch im Quellcode gibt es an der ein oder anderen Stelle andere Möglichkeiten der Umsetzung. Gerade die Aktualisierung der importierten Daten könnte anders realisierbar sein.

Eine Veröffentlichung des Plugins käme in Frage, allerdings müsste dazu viel angepasst werden, da momentan alles auf die Seite konzertkalender.net zugeschnitten ist. Es könnte darüber hinaus eine spezielle Verwaltungs- oder Einstellungsseite angelegt werden, bei der z.B. die gewünschten Eigenschaften einer Veranstaltung, die verglichen werden sollen, auswählbar sind. So könnte das Plugin individuell anpassbar sein.

Das Hinzufügen neuer Vergleichsinformationen oder Neuerungen an der Tabelle sind ebenfalls denkbar. Gerade, als Affiliate-Seite wären z. B. noch Links zu den Veranstaltungsseiten oder dem Ticketverkauf sinnvoll. Momentan stehen acht Auswahlfelder zum Vergleich zur Verfügung. Diese könnten so umgebaut werden, dass per Klick neue hinzugefügt oder vorhandene entfernt werden können. Des Weiteren gilt es den in Abschnitt 6.2 beschriebenen Fehler zu beheben. Es besteht demnach noch Freiraum zur Weiterentwicklung des Plugins im Rahmen künftiger Forschungsarbeiten.

Alles in allem lässt sich jedoch sagen, dass Plugins eine sinnvolle Ergänzung darstellen, die durch die Automatisierung der Vorgänge Aufwand und Zeit einsparen können. So sind die Erkenntnisse dieser Forschungsarbeit ein wichtiger Beitrag, die vielseitigen Anwendungen von Plugins weiter zu entwickeln und anzuwenden.

Literaturverzeichnis

Achour, Medi u.a. (2020): PHP-Handbuch - Manual. Zuletzt aktualisiert am 03.06.2020
Online unter <https://www.php.net/manual/de/index.php> [Abruf am 04.06.2020]

Adams, Ralf (2020): SQL Der Grundkurs für Ausbildung und Praxis: Mit Beispielen in MySQL/MariaDB, PostgreSQL und T-SQL. 3., akt. Auflage. München: Carl Hanser Verlag.
Online unter DOI <https://doi.org/10.3139/9783446462748> [Abruf am 06.05.2020]

Bondari, Brian; Griffiths, Everett (2011): WordPress 3 Plugin Development Essentials. Create your own powerful, interactive plugins to extend and add features to your WordPress site. 1.Auflage. Birmingham: Packt Publishing Ltd

Bühler, Peter, Schlaich, Patrick, Sinner, Dominik (2017): HTML5 und CSS3: Semantik-Design-Responsive-Layouts. 1.Auflage. Berlin, Heidelberg: Springer Vieweg. Online unter DOI <https://doi.org/10.1007/978-3-662-53916-3>

Endcore GmbH (o.J.): Das WordPress Affiliate Theme. Online unter <https://affiliate-theme.io/> [Abruf am 11.05.2020]

Fuchs, Elmar (2018): SQL Grundlagen und Datenbankdesign. 5. Ausgabe. Bodenheim: Herdt-Verlag für Bildungsmedien (Hochschulversion). Online unter <https://herdt-campus.com/product/SQL> [Abruf am 06.05.2020]

Google (2020): Google Suche: Huntenburger Open Air Festival. Online unter <https://www.google.com/search?client=firefox-b-d&q=Huntenburger+Open+Air+Festival> [Abruf am 14.05.2020]

Grochtdreis, Jens (2014): Tabellen die sich anpassen. In: Webkrauts. Blogeintrag vom 21.12.2014. Online unter <http://webkrauts.de/artikel/2014/responsive-tabellen> [Abruf am 05.06.2020]

Grochtdreis, Jens (o.J.): Table Demo 4 - responsiv. Online unter <https://codepen.io/jens-gro/pen/zxYOMY> [Abruf am 05.06.2020]

Haunschild, Marc (2017): CSS Cascading Style Sheet: Level 3 Grundlagen. 3.Ausgabe. Bodenheim: Herdt-Verlag für Bildungsmedien (Hochschulversion). Online unter <https://herdt-campus.com/product/CSS3> [Abruf am 05.05.2020]

Heller, Stephan (2016): PHP 7.0 – Dynamische Webseiten erstellen. 1.Ausgabe. Bodenheim: Herdt-Verlag für Bildungsmedien (Hochschulversion). Online unter <https://herdt-campus.com/product/GPHP7> [Abruf am 05.05.2020]

Kommer, Isolde (2019): WordPress 5 Grundlagen. 1. Ausgabe. Bodenheim: Herdt-Verlag für Bildungsmedien (Hochschulversion). Online unter <https://herdt-campus.com/product/WORDPR5> [Abruf am 04.05.2020]

Konzertkalender.net (2020a): konzertkalender.net: Startseite. Online unter <https://www.konzertkalender.net/> [Abruf am 14.05.2020]

Konzertkalender.net (2020b): konzertkalender.net: Rock im Park. Online unter <https://www.konzertkalender.net/veranstaltungen/rock-im-park/> [Abruf am 14.05.2020]

Konzertkalender.net (2020c): konzertkalender.net: Festivalvergleich. Online unter <https://www.konzertkalender.net/vergleich/> [Abruf am 08.06.2020]

Lammenett, Erwin (2019): Praxiswissen Online-Marketing. Affiliate-, Influencer-, Content- und E-Mail-Marketing, Google Ads, SEO, Social Media, Online- inklusive Facebook-Werbung. 7., überarb. u. erw. Auflage. Wiesbaden: Springer Gabler

Lefebvre, Yannick (2012): WordPress Plugin Development Cookbook. Over 80 step-by-step recipes to extend the most popular CMS and share your creations with its community. 1.Auflage. Birmingham: Packt Publishing Ltd.

Lewandowski, Dirk (2018): Suchmaschinen verstehen. 2 überarb. und erw. Auflage. Berlin, Heidelberg: Springer Vieweg. Online unter <http://dx.doi.org/10.1007/978-3-662-56411-0> [Abruf am 15.05.2020]

Pixelite (2020a): Event Manager for WordPress: About us. Online unter <https://wp-events-plugin.com/about/> [Abruf am 13.05.2020]

Pixelite (2020b): Event Manager for WordPress: Features. Online unter <https://wp-events-plugin.com/features/> [Abruf am 13.05.2020]

Schema.org (o.J.): Home – schema.org. Online unter <https://schema.org/> [Abruf am 14.05.2020]

Schicker, Edwin (2017): Datenbanken und SQL: Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL. 5., akt. u. erw. Auflage. Wiesbaden: Springer Vieweg. Online unter DOI <http://dx.doi.org/10.1007/978-3-658-16129-3> [Abruf am 06.05.2020]

Schröder, Heiko (2012): HTML 4: Webseiten gestalten. 5., akt. Ausgabe. Bodenheim: Herdt-Verlag für Bildungsmedien (Hochschulversion). Online unter <https://herdt-campus.com/product/HTML4F> [Abruf am 05.05.2020]

Starr, Jeff (2020): WordPress uninstall.php file – The Complete Guide. In: Digging Into WordPress. Blogbeitrag vom 11.03.2020. Online unter <https://digwp.com/2019/11/wordpress-uninstall-php/> [Abruf am 04.06.2020]

Statista.com (2020): Ranking der 10 Content-Management-Systeme (CMS) weltweit nach Marktanteil im Januar 2020, unter: <https://de.statista.com/statistik/daten/studie/320670/umfrage/marktanteile-der-content-management-systeme-cms-weltweit/> [Abruf am 15.01.2020]

Steyer, Ralph (2019): Objektorientierte Programmierung mit PHP 7. 1. Ausgabe. Bodenheim: Herdt-Verlag für Bildungsmedien (Hochschulversion). Online unter <https://herdt-campus.com/product/OOP-PHP7> [Abruf am 04.05.2020]

Steyer, Ralph (2018): jQuery: Das universelle JavaScript-Framework für das interaktive Web und mobile Apps. 2., akt. Auflage. München: Carl Hanser Verlag. Online unter <https://doi.org/10.3139/9783446456518> [Abruf am 15.05.2020]

Steyer, Ralph (2016): WordPress: Einführung in das Content Management System. 1. Auflage. Wiesbaden: Springer

WordPress.org (o.J.a): Codex. Register post type. Online unter https://codex.wordpress.org/Function_Reference/register_post_type [Abruf am 12.06.2020]

WordPress.org (o.J.b): Codex. Template redirect. Online unter https://codex.wordpress.org/Plugin_API/Action_Reference/template_redirect [Abruf am 14.06.2020]

WordPress.org (o.J.c): Code Reference. wp_enqueue_style(). Online unter https://codex.wordpress.org/Plugin_API/Action_Reference/template_redirect [Abruf am 14.06.2020]

Yoast BV (2020): Yoast SEO für everyone. Yoast SEO for WordPress. Online unter <https://yoast.com/wordpress/plugins/seo/> [Abruf am 15.05.2020]

Eidesstattliche Erklärung

„Hiermit erkläre ich an Eides statt, dass ich die eingereichte Bachelorarbeit selbständig und ohne fremde Hilfe verfasst, keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.“

Anhang 1: index.php

```
<?php
/*
  Plugin Name: WP-Veranstungsvergleichsplugin
  Plugin URI:
  Description: Ein Plugin zum Vergleich von Veranstaltungen.
  Importiere bereits vorhanden Veranstaltungen aus dem Events Manager,
  um sie miteinander zu vergleichen oder füge eigene Veranstaltungen der Ta-
  belle hinzu.
  Version: 1.0
  Author: Lukas Fischer
  Author URI: https://www.konzertkalender.net/
  */
defined('ABSPATH') or die("Thanks for visting");
// Einbindung erforderlicher Daten
require_once ('includes/importdaten.php');
require_once ('includes/tabelle.php');
require_once ('includes/startseite.php');

// Aufruf der Hooks und Funktionen

// Shortcode
add_shortcode( 'vergleichstabelle', 'erzeuge_tabelle' );

// Fügt Menü ein
add_action('admin_menu', 'menu');
// Funktion für das Menüpunkt
function menu(){
    add_menu_page( 'Festivalvergleich',
                  'Vergleichstabelle',
                  'manage_options',
                  'vergleich_plugin',
                  'anleitung',
                  plugins_url( 'images/konzertkalender.png', __FILE__ )
    );
    add_submenu_page('vergleich_plugin',
                    'Importfunktion',
                    'Import',
                    'manage_options',
                    'vergleich_plugin_import',
                    'importdata' );

    add_submenu_page('vergleich_plugin',
                    'Eigene Einträge',
                    'Eigene Einträge',
                    'manage_options',
                    'edit.php?post_type=veranstaltung');
}
// fügt CPT hinzu
add_action( 'init', 'create_post_type' );
// Funktion zum CPT
function create_post_type() {
    register_post_type( 'veranstaltung',
        array(
            'labels' => array(
                'name' => 'Einträge',
                'singular_name' => 'Eintrag',
                'add_new' => 'Eintrag hinzufügen',
            ),
        ),
    );
}
```

```

        'add_new_item' => 'Neuen Tabellen Eintrag anlegen',
        'edit' => 'Bearbeiten',
        'edit_item' => 'Eintrag bearbeiten',
        'new_item' => 'Neuer Eintrag',
        'view' => 'Ansehen',
        'view_item' => 'Ansehen',
        'search_items' => 'Suche',
        'not_found' => 'Keine Einträge gefunden',
        'not_found_in_trash' => 'Kein Ergebnis',
    ),
    'public' => true, //
    'supports' => array( 'title' ),
    'exclude_from_search' => true,
    'show_in_menu' => 'edit.php?post_type=veranstaltung',
)
);
}
// fügt Metabox ein
add_action( 'admin_init', 'metabox' );
// Funktion für die Metabox
function metabox() {
    add_meta_box( 'details_meta_box',
        'Informationen',
        'meta_box_details',
        'veranstaltung',
        'normal',
        'high' );
}
//Funktion für Inhalt und Details der Metabox
function meta_box_details( $veranstaltung ) {
    $start = esc_html( get_post_meta( $veranstaltung->ID, 'beginn', true ) );
    $ende = esc_html( get_post_meta( $veranstaltung->ID, 'ende', true ) );
    $camping = esc_html( get_post_meta( $veranstaltung->ID, 'campingplatz',
true ) );
    $preis = esc_html( get_post_meta( $veranstaltung->ID, 'preis_festival',
true ) );
    $bandanzahl = esc_html( get_post_meta( $veranstaltung->ID, 'anzahl_band',
true ) );
    $besucheranzahl = esc_html( get_post_meta( $veranstaltung->ID, 'anzahl_besucher', true ) );
    ?>
    <table>
        <tr>
            <td style="width: 100%">Von</td>
            <td><input type="date" size="80" name="startdatum" value="<?php
echo $start; ?>" /></td>
        </tr>
        <tr>
            <td style="width: 100%">Bis</td>
            <td><input type="date" size="80" name="enddatum" value="<?php
echo $ende; ?>" /></td>
        </tr>
        <tr>
            <td style="width: 100%">Preis</td>
            <td><input type="number" size="80" name="kosten" value="<?php
echo $preis; ?>" /></td>
        </tr>
        <tr>
            <td style="width: 100%">Anzahl Bands</td>
            <td><input type="number" size="80" name="bands" value="<?php echo
$bandanzahl; ?>" />

```



```

        </tr>
        <tr>
            <td style="width: 100%">Besucher</td>
            <td><input type="number" size="80" name="besucher" value="<?php
echo $besucheranzahl; ?>" />
        </tr>
        <tr>
            <td style="width: 100%">Camping</td>
            <td><input type="text" size="80" name="camping_moglich"
value="<?php echo $camping; ?>" /></td>
        </tr>
    </table>
    <?php }

// Speicherfunktion für die Metabox
add_action( 'save_post', 'save_info', 10, 2 );
function save_info( $veranstaltung_id, $veranstaltung ) {
    if ( $veranstaltung->post_type == 'veranstaltung' ) {
        if ( isset( $_POST['camping_moglich'] ) && $_POST['camp-
ing_moglich'] != '' ) {
            update_post_meta( $veranstaltung_id, 'campingplatz',
$_POST['camping_moglich'] );
        }
        if ( isset( $_POST['bands'] ) && $_POST['bands'] != '' ) {
            update_post_meta( $veranstaltung_id, 'anzahl_band',
$_POST['bands'] );
        }
        if ( isset( $_POST['kosten'] ) && $_POST['kosten'] != '' ) {
            update_post_meta( $veranstaltung_id, 'preis_festival',
$_POST['kosten'] );
        }
        if ( isset( $_POST['besucher'] ) && $_POST['besucher'] != ''
) {
            update_post_meta( $veranstaltung_id,
'anzahl_besucher', $_POST['besucher'] );
        }
        if ( isset( $_POST['startdatum'] ) && $_POST['startdatum'] !=
'' ) {
            update_post_meta( $veranstaltung_id, 'beginn',
$_POST['startdatum'] );
        }
        if ( isset( $_POST['enddatum'] ) && $_POST['enddatum'] != ''
) {
            update_post_meta( $veranstaltung_id, 'ende',
$_POST['enddatum'] );
        }
    }
}

// Einbinden der CSS-Datei
add_action( 'template_redirect', 'tabellecss' );
// Funktion für das Einbinden der CSS-Datei
function tabellecss() {
    if ( is_page( 'vergleich' ) ) {
        wp_enqueue_style( 'style-name', plugins_url('/css/ta-
belle.css', __FILE__));
    }
}
?>

```

Anhang 2: uninstall.php

```
<?php
//Aufruf, wenn Plugin über das Dashboard deinstalliert wird
if ( !defined( 'WP_UNINSTALL_PLUGIN' ) ) {
    exit();
}

global $wpdb;

$wpdb->query( "DROP TABLE IF EXISTS vergleich" );
$wpdb->query( "DELETE FROM datenbankname_posts96 WHERE post_type ='veranstal-
tung'" );
?>
```

Anhang 3: ausgeben.php

```
<?php
/*Ausgeben bekommt bei Aufruf SQL -Query übergeben. Packt diese dann in ein
Array und gibt eine Liste mit den Namen der Elemente aus.
Ausgabe im Back-End beim importieren*/

function ausgeben($statement){
    if ($statement != NULL) {
        $zaehler = 0;
        while ($reihe = $statement->fetch_row()) {
            $festival[] = array($reihe[0]);
            $zaehler++;
        }
        echo $zaehler." Festivals importiert<br><br>";
        for($i= 0; $i < $zaehler; $i++){
            echo $festival[$i][0];
            if($i < $zaehler-1){
                echo ", ";
            }
        }
    } else {
        echo "<p>Keine Daten in der Datenbank vorhanden</p>";
    }
}
?>
```

⁹⁶ Name der Datenbank geändert in datenbankname

Anhang 4: connection.php

```
<?php
// Verbindungs-Objekt mit Zugangsdaten
// Stellt Verbindung zur Datenbank her
$db = new mysqli(██████████, ██████████, ██████████, ██████████);97

$db->set_charset("utf8");

if (mysqli_connect_errno()) {
    printf("Verbindung fehlgeschlagen: %s\n", mysqli_connect_error());
    exit();
}

?>
```

Anhang 5: datumformat.php

```
<?php
// wandelt Datumformat um
// nutzt Format 2020-01-15
function format($date){
    $d = substr("$date", -2); // speichert letzten zwei D = 15
    $m = substr("$date", -5,2); // speichert ab den 5 letzten Wert zwei
    Zeichen M = 01
    $y = substr("$date", -8,2); // speichert ab den 5 letzten Wert zwei
    Zeichen j = 20
    $datum = '&nbsp;'.$d.'.'.$m.'.'.$y.'&nbsp;'; // Ausgabe des Datumsformat
    mit Anpassung an das Plugin
    return $datum;
}

?>
```

Anhang 6: dbtabelle.php

```
<?php
// Prüft ob die Tabelle der Datenbank vorhanden ist und erstellt diese falls
keine vorhanden ist
$sqltabelle = "SELECT * FROM vergleich";
$ergebnis = $db->query($sqltabelle);
if($ergebnis){
    echo "Plugin ist f&uuml;r den Import bereit.";
} else{
    $neuetabelle = "CREATE TABLE vergleich (name VARCHAR (200), sdatum DATE
NOT NULL, edatum DATE NOT NULL, preis FLOAT, camping VARCHAR (100), bands IN-
TEGER, besucher VARCHAR (100), import VARCHAR (100) NOT NULL, id int, PRIMARY
KEY(name) ) ";
    $db->query($neuetabelle);
    echo "Plugin Einrichtung abgeschlossen.";
}

?>
```

⁹⁷ Verbindungsinformationen durch Platzhalter ersetzt

Anhang 7: importdaten.php

```
<?php
/* Importfunktion
 * Auslesen und kopieren von Veranstaltungen aus einer Datenbanktabelle in
 die Vergleichsdatenbanktabelle
 */
function importdata(){
    // Einbindung anderer Dateien
    require_once ('connection.php');
    require_once ('dbtabelle.php');
    require_once ('ausgeben.php');
    ?>
    <h1>Import</h1>
    <form action='' method='POST'>
        <h2>Import aus Veranstaltungsdatenbank</h2>
        <input type="submit" value="Veranstaltungen importieren" name="import"
 />
        <input type="submit" value="Importierte Veranstaltung entfernen"
 name="Loeschen" /><br/>
        <h2>Import aus eigenen Veranstaltungen</h2>
        <input type="submit" value="Eigenen Veranstaltungen Importieren"
 name="importeigene" />
        <input type="submit" value="Eigenen Veranstaltungen entfernen"
 name="Loescheneigenen" /><br/>
        <h2>Aktualisieren</h2>
        <input type="submit" value="Eintr&auml;ge aktualisieren" name="update"
 /> <br/>
    </form>
    <?php
    $inhalt;
    $inhaltqown;
    // Benötigte SQL-Befehle
    //Einfügen von Veranstaltungen aus der Eventdatenbanktabelle
    $sqlimport = "INSERT INTO vergleich (id, name, sdatum, edatum)
        SELECT post_id, event_name, event_start_date, event_end_date
        FROM datenbankname_em_events98
        LEFT JOIN datenbankname_posts
        ON datenbankname_em_events.post_id = datenbankname_posts.ID
        WHERE post_status = 'publish'";
    //Einfügen von eigenen Veranstaltungen in die Eventdatenbanktabelle
    $sqlleigen = "INSERT INTO vergleich (id, name, import) SELECT datenbank-
name_posts.ID,post_title,post_type FROM datenbankname_posts WHERE post_type =
'veranstaltung' AND (post_status = 'draft' OR post_status = 'publish')";
    // Aktualisierung von Veranstaltungen aus der Eventdatenbanktabelle
    $sqlbesucher = "UPDATE vergleich SET vergleich.besucher = (SELECT daten-
bankname_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'Zuschau-
er') WHERE `import` <>'veranstaltung'";
    $sqlbands = "UPDATE vergleich SET vergleich.bands = (SELECT datenbank-
name_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id = da-
tenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'An-
zahl_Bands') WHERE `import` <>'veranstaltung'";
    $sqlcamping = "UPDATE vergleich SET vergleich.camping = (SELECT daten-
bankname_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'Cam-
ping') WHERE `import` <>'veranstaltung'";
```

⁹⁸ Name der Datenbank im folgenden Code geändert in datenbankname

```

    $sqlpreis = "UPDATE vergleich SET vergleich.preis = (SELECT datenbank-
name_.meta_value FROM datenbankname_postmeta WHERE vergleich.id = datenbank-
name_postmeta.post_id AND datenbankname_postmeta.meta_key = 'Eintritt') WHERE
`import` <>'veranstaltung'";
// Aktualisierung von eigenen Veranstaltungen
    $sqlsdatum = "UPDATE vergleich SET vergleich.sdatum = (SELECT datenbank-
name_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'beginn')
WHERE `import` = 'veranstaltung'";
    $sqledatum = "UPDATE vergleich SET vergleich.edatum = (SELECT datenbank-
name_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'ende')
WHERE `import` = 'veranstaltung'";
    $sqlkartenpreis = "UPDATE vergleich SET vergleich.preis = (SELECT daten-
bankname_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key
='preis_festival') WHERE `import` = 'veranstaltung'";
    $sqlcamp = "UPDATE vergleich SET vergleich.camping = (SELECT datenbank-
name_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key = 'camping-
platz') WHERE `import` = 'veranstaltung'";
    $sqlband = "UPDATE vergleich SET vergleich.bands = (SELECT datenbank-
name_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key
='anzahl_band') WHERE `import` = 'veranstaltung'";
    $sqlzuschauer = "UPDATE vergleich SET vergleich.besucher = (SELECT daten-
bankname_postmeta.meta_value FROM datenbankname_postmeta WHERE vergleich.id =
datenbankname_postmeta.post_id AND datenbankname_postmeta.meta_key
='anzahl_besucher') WHERE `import` = 'veranstaltung'";
    // Löschen von Veranstaltungen aus Vergleichstabelle, die aus der Event-
datenbanktabelle stammen
    $sqldelete = "DELETE FROM `vergleich` WHERE `import` <>'veranstaltung'";
    // Löschen von eigenen Veranstaltungen aus Vergleichstabelle
    $sqldeleteown = "DELETE FROM `vergleich` WHERE `import` = 'veranstal-
tung'";
    // Löschen alle Eintragungen
    $sqldeleteall = "DELETE FROM `vergleich`";
    // Ausgeben von Veranstaltungen, die aus der Eventdatenbank stammen
    $sqlauswahlimport = "SELECT `name` FROM `vergleich` WHERE `import`
<>'veranstaltung' ORDER BY `name` ASC";
    // Ausgeben von eigenen Veranstaltungen
    $sqlauswahleigen = "SELECT `name` FROM `vergleich` WHERE `import` =
'veranstaltung' ORDER BY `name` ASC";
// Zählung von Veranstaltungen in Datenbank-Tabelle
    $sqldatacheck = $db->query("SELECT COUNT(*) FROM vergleich WHERE `import`
<>'veranstaltung'");
    $sqldatacheckown = $db->query("SELECT COUNT(*) FROM vergleich WHERE `im-
port` = 'veranstaltung'");
// Import
    if(isset($_POST['import'])){
        $db->query($sqlimport);
        $db->query($sqlpreis);
        $db->query($sqlbesucher);
        $db->query($sqlbands);
        $db->query($sqlcamping);
        echo "<br>Der Import war erfolgreich.<br>";
    } else if(isset($_POST['loeschen'])) {
        $db->query($sqldelete);
        echo "<br>Daten gel&ouml;sch<br>";
    } else if(isset($_POST['importeigene'])) {
        $db->query($sqleigen);
    }

```

```

$db->query($sqlsdatum);
$db->query($sqledatum);
$db->query($sqlkartenpreis);
$db->query($sqlcamp);
$db->query($sqlband);
$db->query($sqlzuschauer);
echo "<br>Eigene Festivals wurden importiert.<br>";
} else if(isset($_POST['loescheneigenen'])) {
    $db->query($sqldeleteown);
    echo "<br>Eigene Festivals wurden gel&ouml;scht.<br>";
} else if(isset($_POST['update'])) {
    while ($reihe = $sqldatacheck->fetch_row()) {
        $countimport[] = array($reihe[0]);
    }
    while ($reihe = $sqldatacheckown->fetch_row()) {
        $countimportown[] = array($reihe[0]);
    }
    $anzahlimport = $countimport[0][0];
    $anzahlimportown = $countimportown[0][0];
    if ($countimport[0][0]>0){
        $db->query($sqldelete);
        $db->query($sqlimport);
        $db->query($sqlpreis);
        $db->query($sqlbesucher);
        $db->query($sqlbands);
        $db->query($sqlcamping);
        echo "Eintr&auml;ge aus Events-Datenbanktabelle aktu-
alisiert.<br>";
    } else if ($anzahlimport==0){
        echo "Keine Eintr&auml;ge aus Events-Datenbanktabelle
vorhanden.<br>";
    } if ($countimportown[0][0]>0){
        $db->query($sqldeleteown);
        $db->query($sqleigen);
        $db->query($sqlsdatum);
        $db->query($sqledatum);
        $db->query($sqlkartenpreis);
        $db->query($sqlcamp);
        $db->query($sqlband);
        $db->query($sqlzuschauer);
        echo "Eigene Eintr&auml;ge aktualisiert.<br>";
    } else if ($anzahlimportown==0){
        echo "Keine eigenen Eintr&auml;ge vorhanden.<br>";
    }
}
// Aufruf der ausgabe-Funktion
echo "<br><h2>Liste der eigenen Festivals:</h2>";
ausgeben($db->query($sqlauswahleigen));
echo "<br><h2>Liste der importierten Festivals aus Events-Datenbankta-
belle:</h2>";
ausgeben($db->query($sqlauswahlimport));
}??>

```

Anhang 8: startseite.php

```
<?php
// Anleitung und Startseite im Menü
function anleitung(){?>
    <div>
        <h1>Wie funktioniert das Plugin?</h1>
        <p>Mit dem Plugin können Veranstaltungen, aus der Datenbank im-
portiert
        und in eine Tabelle eingefügt und verglichen werden.</p>
        <h3>Menüpunkt Import:</h3>
        <ul>
            <li>&#8226; <input type="submit" value="Daten-Importieren"
name="" /> </li> Importieren von Veranstaltungen aus der Eventdatenbank oder
eigener Veranstaltungen.</li>
            <li>&#8226; <input type="submit" value="Importierte Veranstaltung
entfernen" name="" /><br/> Löschen von Veranstaltungen aus der Event-
datenbank oder eigener Veranstaltungen.</li>
            <li>&#8226; <input type="submit" value="Einträge aktu-
alisieren" name="" /> <br/> Aktualisieren der Einträge nach dem
Änderungen durchgeführt wurden. </li>
        </ul>

        <h3>Menüpunkt Eigene Einträge:</h3>
        <ul>
            <li>&#8226; Eintragen, Löschen und Bearbeiten von
eigenen Veranstaltungen, die nicht auf der Seite gelistet sind.</li>
            <li>&#8226; Events die veröffentlicht sind bekommen
eine aufrufbare URL, und können vom Nutzer aufgerufen werden. Soll diese
Seite nicht sichtbar sein, muss der Eintrag auf Entwurf gestellt werden.</li>
        </ul>

        <h3>Tabelle auf Seite oder Beitrag sichtbar machen:</h3>
        <ul>
            <li>&#8226; durch den Shortcode [vergleichstabelle] wird
die Tabelle auf einer Seite oder einen Beitrag sichtbar</li>
        </ul>
    </div>
<?php }?>
```

Anhang 9: tabelle.php

```
<?php
// erstellt die Vergleichstabelle
function erzeugeTabelle(){
// Einbindung anderer Dateien
    require_once ('connection.php');
    require_once ('tage.php');
    require_once ('datumformat.php');

$sql = "Select * from vergleich ORDER BY `name` ASC";
$zaehler = 0;

// schreibt den Inhalt der DB-Tabelle in ein Array
if ($r = $db->query($sql)) {
    while ($reihe = $r->fetch_row()) {
        $festival[] = array($reihe[0], $reihe[1], $reihe[2], $reihe[3].
&euro;", $reihe[4], $reihe[5], $reihe[6], $reihe[7]);
        $zaehler++;
    }
}

// Tabelle und Formular zum Auswählen der im Array gespeicherten Aus-
wählmöglichkeiten
?>
<form action="" method="POST">
<table>
    <thead>
    <tr>
    <th>Festivalname</th>
    <th>Start</th>
    <th>Ende</th>
    <th>&nbsp;Tage&nbsp;</th>
    <th>&nbsp;Preis&nbsp;</th>
    <th>&nbsp;Camping&nbsp;</th>
    <th>&nbsp;Bands&nbsp;</th>
    <th>&nbsp;Besucher&nbsp;</th>
    </tr>
    </thead>
    <tbody>
    <tr>
    <td data-title="Festivalname">
        <select name="festival1" id="fest1">
            <option selected="selected" disabled="disabled"><?php echo
$festival[$_POST['festival1']][0]?></option>
            <?php
                for($i= 0; $i < $zaehler; $i++){
                    ?><option value="<?php echo $i ?>"><?php echo $festi-
val[$i][0] ?></option>
                <?php } ?>
            </select>
        </td>
    <td data-title="Start">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo format($festival[$_POST['festival1']][1]);
            }
        </td>
    </tr>
    </tbody>
</table>
</form>
```



```

        ?>
    </td>
    <td data-title="Ende">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo format($festival[$_POST['festival1']][2]);
            }
        ?>
    </td>
    <td data-title="Tage">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo tage($festival[$_POST['festival1']][1],$festi-
val[$_POST['festival1']][2]);
            }
        ?>
    </td>
    <td data-title="Preis">
        <?php
            echo $festival[$_POST['festival1']][3];
        ?>
    </td>
    <td data-title="Camping">
        <?php
            echo $festival[$_POST['festival1']][4];
        ?>
    </td>
    <td data-title="Bands">
        <?php
            echo $festival[$_POST['festival1']][5];
        ?>
    </td>
    <td data-title="Besucher">
        <?php
            echo $festival[$_POST['festival1']][6];
        ?>
    </td>
</tr>
<tr>
    <td data-title="Festivalname">
        <select name="festival2" id="fest2">
            <option selected="selected" disabled="disabled"><?php echo
$festival[$_POST['festival2']][0]?></option>
            <?php
                for($i= 0; $i < $zaehler; $i++){
                    ?><option value="<?php echo $i ?>"><?php echo $festi-
val[$i][0] ?></option>
                <?php } ?>
            </select>
        </td>

    <td data-title="Start">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo format($festival[$_POST['festival2']][1]);
            }
        ?>
    </td>
    <td data-title="Ende">
        <?php
            if(isset($_POST['auswaehlen'])){

```

```

        echo format($festival[$_POST['festival2']][2]);
    }
    ?>
</td>
<td data-title="Tage">
    <?php
    if(isset($_POST['auswaehlen'])){
        echo tage($festival[$_POST['festival2']][1],$festi-
val[$_POST['festival2']][2]);
    }
    ?>
</td>
<td data-title="Preis">
    <?php
    echo $festival[$_POST['festival2']][3];
    ?>
</td>
<td data-title="Camping">
    <?php
    echo $festival[$_POST['festival2']][4];
    ?>
</td>
<td data-title="Bands">
    <?php
    echo $festival[$_POST['festival2']][5];
    ?>
</td>
<td data-title="Besucher">
    <?php
    echo $festival[$_POST['festival2']][6];
    ?>
</td>
</tr>
<tr>
    <td data-title="Festivalname">
        <select name="festival3" id="fest3">
            <option selected="selected" disabled="disabled"><?php echo
$festival[$_POST['festival3']][0]?></option>
            <?php
            for($i= 0; $i < $zaehler; $i++){
                ?><option value="<?php echo $i ?>"><?php echo $festi-
val[$i][0] ?></option>
            <?php } ?>
            </select>
        </td>

    <td data-title="Start">
        <?php
        if(isset($_POST['auswaehlen'])){
            echo format($festival[$_POST['festival3']][1]);
        }
        ?>
    </td>
    <td data-title="Ende">
        <?php
        if(isset($_POST['auswaehlen'])){
            echo format($festival[$_POST['festival3']][2]);
        }
        ?>
    </td>
    <td data-title="Tage">

```

```

        <?php
        if(isset($_POST['auswaehlen'])){
            echo tage($festival[$_POST['festival3']][1],$festi-
            val[$_POST['festival3']][2]);
        }
        ?>
    </td>
    <td data-title="Preis">
        <?php
        echo $festival[$_POST['festival3']][3];
        ?>
    </td>
    <td data-title="Camping">
        <?php
        echo $festival[$_POST['festival3']][4];
        ?>
    </td>
    <td data-title="Bands">
        <?php
        echo $festival[$_POST['festival3']][5];
        ?>
    </td>
    <td data-title="Besucher">
        <?php
        echo $festival[$_POST['festival3']][6];
        ?>
    </td>
</tr>
<tr>
    <td data-title="Festivalname">
        <select name="festival4" id="fest4">
            <option selected="selected" disabled="disabled"><?php echo
            $festival[$_POST['festival4']][0]?></option>
            <?php
            for($i= 0; $i < $zaehler; $i++){
                ?><option value="<?php echo $i ?>"><?php echo $festi-
                val[$i][0] ?></option>
            <?php } ?>
        </select>
    </td>

    <td data-title="Start">
        <?php
        if(isset($_POST['auswaehlen'])){
            echo format($festival[$_POST['festival4']][1]);
        }
        ?>
    </td>
    <td data-title="Ende">
        <?php
        if(isset($_POST['auswaehlen'])){
            echo format($festival[$_POST['festival4']][2]);
        }
        ?>
    </td>
    <td data-title="Tage">
        <?php
        if(isset($_POST['auswaehlen'])){
            echo tage($festival[$_POST['festival4']][1],$festi-
            val[$_POST['festival4']][2]);
        }

```

```

        ?>
    </td>
    <td data-title="Preis">
        <?php
            echo $festival[$_POST['festival4']][3];
        ?>
    </td>
    <td data-title="Camping">
        <?php
            echo $festival[$_POST['festival4']][4];
        ?>
    </td>
    <td data-title="Bands">
        <?php
            echo $festival[$_POST['festival4']][5];
        ?>
    </td>
    <td data-title="Besucher">
        <?php
            echo $festival[$_POST['festival4']][6];
        ?>
    </td>
</tr>
<tr>
    <td data-title="Festivalname">
        <select name="festival5" id="fest5">
            <option selected="selected" disabled="disabled"><?php echo
$festival[$_POST['festival5']][0]?></option>
            <?php
                for($i= 0; $i < $zaehler; $i++){
                    ?><option value="<?php echo $i ?>"><?php echo $festi-
val[$i][0] ?></option>
                <?php } ?>
            </select>
        </td>

    <td data-title="Start">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo format($festival[$_POST['festival5']][1]);
            }
        ?>
    </td>
    <td data-title="Ende">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo format($festival[$_POST['festival5']][2]);
            }?>
    </td>
    <td data-title="Tage">
        <?php
            if(isset($_POST['auswaehlen'])){
                echo tage($festival[$_POST['festival5']][1],$festi-
val[$_POST['festival5']][2]);
            }?>
    </td>
    <td data-title="Preis">
        <?php
            echo $festival[$_POST['festival5']][3];
        ?>
    </td>

```

```

        <td data-title="Camping">
            <?php
                echo $festival[$_POST['festival5']][4];
            ?>
        </td>
        <td data-title="Bands">
            <?php
                echo $festival[$_POST['festival5']][5];
            ?>
        </td>
        <td data-title="Besucher">
            <?php
                echo $festival[$_POST['festival5']][6];
            ?>
        </td>
    </tr>
    <tr>
        <td data-title="Festivalname">
            <select name="festival6" id="fest6">
                <option selected="selected" disabled="disabled"><?php echo
                    $festival[$_POST['festival6']][0]?></option>
                <?php
                    for($i= 0; $i < $zaehler; $i++){
                        ?><option value="<?php echo $i ?>"><?php echo $festi-
                            val[$i][0] ?></option>
                    } ?>
            </select>
        </td>

        <td data-title="Start">
            <?php
                if(isset($_POST['auswaehlen'])){
                    echo format($festival[$_POST['festival6']][1]);
                }?>
        </td>
        <td data-title="Ende">
            <?php
                if(isset($_POST['auswaehlen'])){
                    echo format($festival[$_POST['festival6']][2]);
                }?>
        </td>
        <td data-title="Tage">
            <?php
                if(isset($_POST['auswaehlen'])){
                    echo tage($festival[$_POST['festival6']][1],$festi-
                        val[$_POST['festival6']][2]);
                }
            ?>
        </td>
        <td data-title="Preis">
            <?php
                echo $festival[$_POST['festival6']][3];
            ?>
        </td>
        <td data-title="Camping">
            <?php
                echo $festival[$_POST['festival6']][4];
            ?>
        </td>
        <td data-title="Bands">
            <?php

```

```

        echo $festival[$_POST['festival6']][5];
    ?>
</td>
<td data-title="Besucher">
    <?php
        echo $festival[$_POST['festival6']][6];
    ?>
</td>
</tr>
<tr>
<td data-title="Festivalname">
    <select name="festival7" id="fest7">
        <option selected="selected" disabled="disabled"><?php echo
$festival[$_POST['festival7']][0]?></option>
        <?php
            for($i= 0; $i < $zaehler; $i++){
                ?><option value="<?php echo $i ?>"><?php echo $festi-
val[$i][0] ?></option>
            <?php } ?>
        </select>
    </td>

<td data-title="Start">
    <?php
        if(isset($_POST['auswaehlen'])){
            echo format($festival[$_POST['festival7']][1]);
        }?>
</td>
<td data-title="Ende">
    <?php
        if(isset($_POST['auswaehlen'])){
            echo format($festival[$_POST['festival7']][2]);
        }?>
</td>
<td data-title="Tage">
    <?php
        if(isset($_POST['auswaehlen'])){
            echo tage($festival[$_POST['festival7']][1],$festi-
val[$_POST['festival7']][2]);
        }?>
</td>
<td data-title="Preis">
    <?php
        echo $festival[$_POST['festival7']][3];
    ?>
</td>
<td data-title="Camping">
    <?php
        echo $festival[$_POST['festival7']][4];
    ?>
</td>
<td data-title="Bands">
    <?php
        echo $festival[$_POST['festival7']][5];
    ?>
</td>
<td data-title="Besucher">
    <?php
        echo $festival[$_POST['festival7']][6];
    ?>
</td>

```

```

</tr>
<tr>
  <td data-title="Festivalname">
    <select name="festival8" id="fest8">
      <option selected="selected" disabled="disabled"><?php echo
$festival[$_POST['festival8']][0]?></option>
      <?php
        for($i= 0; $i < $zaehler; $i++){
          ?><option value="<?php echo $i ?>"><?php echo $festi-
val[$i][0] ?></option>
        <?php } ?>
      </select>
    </td>

    <td data-title="Start">
      <?php
        if(isset($_POST['auswaehlen'])){
          echo format($festival[$_POST['festival8']][1]);
        }?>
    </td>
    <td data-title="Ende">
      <?php
        if(isset($_POST['auswaehlen'])){
          echo format($festival[$_POST['festival8']][2]);
        }?>
    </td>
    <td data-title="Tage">
      <?php
        if(isset($_POST['auswaehlen'])){
          echo tage($festival[$_POST['festival8']][1],$festi-
val[$_POST['festival8']][2]);
        }?>
    </td>
    <td data-title="Preis">
      <?php
        echo $festival[$_POST['festival8']][3];
      ?>
    </td>
    <td data-title="Camping">
      <?php
        echo $festival[$_POST['festival8']][4];
      ?>
    </td>
    <td data-title="Bands">
      <?php
        echo $festival[$_POST['festival8']][5];
      ?>
    </td>
    <td data-title="Besucher">
      <?php
        echo $festival[$_POST['festival8']][6];
      ?>
    </td>
  </tr>
</tbody>
</table>
<input type="submit" name="auswaehlen" value="Festivals ausw&auml;hlen">
</form>
<?php
}??>

```

Anhang 10: tage.php

```
<?php
// Errechnet die Anzahl der Tage zwischen zwei Daten
function tage($starttag, $endtag){
    $enddatum = strtotime($endtag . ' + 1 day'); // Umwandlung in Sekunden
    und Erhöhung um einen Tag
    $datumpluseins = date('Y-m-d', $enddatum); // Umwandlung in Datum Y-M-D
    $datum1 = new DateTime($starttag); // speichern als Datum
    $datum2 = new DateTime($datumpluseins);
    $tage = $datum1->diff($datum2); //Differenz errechnen
    return $tage->format('%a'); // Ausgabe der Tage
}
?>
```

Anhang 11: tabelle.css

```
/* Formatierung der Tabelle und Anpassung auf mobile Geräte */
table {
    width: 100%;
    border-collapse: collapse;
    border-spacing: 0;
    margin: 0;
}

th,td {
    line-height: 2.0em;
    vertical-align: top;
}

th *:first-child,td *:first-child {
    margin-top: 0;
}

thead th {
    text-align: center;
    border-bottom: 3px #126b22 solid;
}

tbody th, td {
    text-align: center;
    border-top: 1px solid #ccc;
}

tbody tr:hover th, tbody tr:hover td {
    background: #f8f8f8;
}

@media only screen and (max-width: 760px), (min-device-width: 768px) and
(max-device-width: 1024px) {
    table, thead, tbody, th, td, tr {
        display: block; /*alle Tabellenzeilen werden untereinander aus-
gegeben */
        text-align: left;
    }
}
```



```

thead { /* Blendet in mobilen Version die Überschriften aus */
    display: none;
}

tr {
    border: 1px solid #ccc;
    margin-bottom: 20px;
}

td {
    border: none;
    position: relative;
}

td:before {
    /* Formatierung der Datenzeilen untereinander */
    content: attr(data-title);
    display: inline-block;
    width: 35%;
    padding-right: 10px;
    white-space: nowrap;
}
}

/* Anpassung für Smartphones */
@media only screen and (min-device-width : 320px) and (max-device-width :
480px) {
    body {
        padding: 0;
        margin: 0;
        width: 320px;
    }

    td {
        padding-left: 45%;
    }

    td:before {
        width: 40%;
    }
}

/* Anpassung für iPads */
@media only screen and (min-device-width: 768px) and (max-device-width:
1024px) {
    body {
        width: 495px;
    }
}
}

```