**Erstellung eines**

**WordPress Plug-In «FAQ»**

FACHARBEIT VON NICOLAS SUTER

BERICHT

**Verfasser**

Name Nicolas Suter

E-Mail Privat nicolas.suter94@hotmail.com

Mobile +41 79 459 07 03

Adresse Privat Eichhölzlistrasse 16

8192 Glattfelden

**Verteiler**

**Hauptexperte**

Name Robert Bonomo

E-Mail Privat robert.bonomo@hispeed.ch

Mobile +41 79 173 17 89

**Nebenexperte**

Name Touseef Arif

E-Mail Privat touseef.arif@gmail.com

Mobile +41 79 903 42 41

**Verantwortliche Fachkraft**

Name Jörn Fischer

E-Mail Privat joern@onebyte.ch

Mobile +46 70 640 18 60

**Spätester Abgabetermin**

Donnerstag, 14.05.2020 12:00

# Inhaltsverzeichnis

[Inhaltsverzeichnis - 3 -](#_Toc38902489)

[Teil 1: Obligatorische Kapitel - 4 -](#_Toc38902490)

[1 Aufgabenstellung - 4 -](#_Toc38902491)

[1.1 Titel der Arbeit *aus PkOrg* - 4 -](#_Toc38902492)

[1.2 Ausgangslage *aus PkOrg* - 4 -](#_Toc38902493)

[1.3 Detaillierte Aufgabenstellung *aus PkOrg* - 4 -](#_Toc38902494)

[2 Mittel und Methoden - 6 -](#_Toc38902495)

[2.1 Projektmethode *aus PkOrg* - 6 -](#_Toc38902496)

[2.2 Zu verwendende Mittel und Methoden *aus PkOrg* - 6 -](#_Toc38902497)

[3 Vorkenntnisse - 7 -](#_Toc38902498)

[3.1 Vorkenntnisse *aus PkOrg* - 7 -](#_Toc38902499)

[3.2 Neue Lerninhalte *aus PkOrg* - 7 -](#_Toc38902500)

[3.2 Vorarbeiten *aus PkOrg* - 7 -](#_Toc38902501)

[3.3 Neue Lerninhalte *aus PkOrg* - 7 -](#_Toc38902502)

[3.4 Arbeiten in den letzten 6 Monaten *aus PkOrg* - 7 -](#_Toc38902503)

[3.6 Erfahrungen mit Wordpress - 8 -](#_Toc38902504)

[4 Organisation - 8 -](#_Toc38902505)

[4.1 Onebyte AG - 8 -](#_Toc38902506)

[4.2 Projektablauforganisation - 8 -](#_Toc38902507)

[5 Zeitplanung - 9 -](#_Toc38902508)

[7 Tagesjournale - 10 -](#_Toc38902509)

# Teil 1: Obligatorische Kapitel

## 1 Aufgabenstellung

Die Aufgabenstellung wurde aus PkOrg übernommen ohne Änderungen daran vorzunehmen. Die Darstellung wurde angepasst und die Rechtschreibprüfung wurde durch Word durchgeführt.

### 1.1 Titel der Arbeit *aus PkOrg*

WordPress Plug-In "FAQ"

### 1.2 Ausgangslage *aus PkOrg*

Für unsere Kunden erstellen wir Webseiten basierend auf dem WordPress CMS nach dem aktuellsten Standard und zu verschiedenen Zwecken. Dabei wird regelmässig von unseren Kunden die Darstellung von FAQs gewünscht, die wir dann manuell erstellen und die auf keiner bestimmten einheitlichen Vorlage basieren. Aus diesen Gründen besteht der Wunsch für ein FAQ Plug-In, welches einfach mittels "Plug-In hinzufügen"-Funktion installiert und abgefüllt werden kann. Dies spart uns beim Aufbau viel Zeit und ist für den Kunden einfacher zu pflegen.

### 1.3 Detaillierte Aufgabenstellung *aus PkOrg*

**Basis und Codeanforderungen**

In WordPress 5.3.x soll ein Custom Post Type Plug-In «FAQ» erstellt werden, welches in jedem Wordpress integriert werden kann. Das Modul basierend auf PHP muss kompatibel sein mit der PHP Version 7.3 und mit HTML 5 und CSS 3 gestaltet werden. Das Plug-In muss nach dem Standard von WordPress Codex mit Berücksichtigung der Mehrsprachigkeit programmiert sein, so dass die FAQ Inhalte über die Verwendung eines dritten Mehrsprachen-Plug-In übersetzt werden kann.

**Funktionelle Grobanforderungen**

Das FAQ Plug-In muss für den Kunden folgende Möglichkeiten bieten:

- Erfassung von einer oder mehreren FAQs im Backend

- Ausgabe der FAQs im Frontend mittels Shortcode

- Eingabe im Backend auf einer beliebigen WordPress Page oder Post

**Detaillierte Anforderung**

**>> Backend / CMS**

Das Plug-In muss über die standardmässige Plug-In Installationsfunktion von WordPress installierbar und aktivierbar sein. Es erhält nach Aktivierung in der linken Admin-Sidebar einen eigenen Menüpunkt, der für sinnvolle Benutzerrollen sichtbar sein soll. Dazu soll ein kleines Konzept genannt werden, worin definiert ist, welche Benutzerrollen (aus dem WordPress Standard) Zugriff auf das Plug-In erhalten sollen.

**>> Backend: Funktionalität**

Eine FAQ sollte folgende Pflicht- sowie Optionale Felder haben:

**Pflichtfelder:**

- Frage (Mehrzeiliges Textfeld)

- Antwort (Mehrzeiliges Textfeld)

**Optionale Felder:**

- Autor (Einzeiliges Textfeld) Wird das optionale Autor Feld leer gelassen, wird automatisch der Beitragsautor als Ausgabewert übermittelt.

Über die Standardfunktion von WordPress «Beitragsbild» soll der Benutzer die Möglichkeit haben, ein Autor-Bild zu einer FAQ hochzuladen.

Beim Abspeichern der Daten wird validiert, ob alle Pflichtfelder ausgefüllt wurden. Falls nicht alle Pflichtfelder ausgefüllt wurden, soll eine entsprechende Meldung erscheinen. Über die Standardfunktion von WordPress soll der Benutzer der FAQ eine Kategorie zuweisen können, so dass im Shortcode nach Kategorie die FAQ ausgegeben werden kann. Die Reihenfolge der ausgegebenen FAQs soll durch den Administrator manuell bestimmt werden können. Die Funktion dieser Bestimmung muss nicht direkt implementiert werden und darf mittels eines dritten Plug-Ins erfolgen, sofern bei der FAQ- Plug-In Installation durch den Benutzer ein entsprechender Hinweis zu einem zusätzlich erforderlichen Plug-In erscheint.

**>> Frontend: Design**

Da unsere Kundenwebseiten individuell auf Kundenvorgaben gestaltet werden, bestehen in diesem Teil der Arbeit keine designspezifischen Vorgaben, ausser dass eine zeitgemässe, responsive Gestaltung der ausgegebenen FAQs zu gewährleisten ist. Dazu kann ein Framework verwendet werden, welches durch den Lernenden evaluiert wird.

**>> Frontend: Funktionalität**

Es muss die Möglichkeit bestehen, auf einer beliebigen Page oder in einem beliebigen Post eine gewünschte FAQ Kategorie mittels Shortcode auszugeben. Es sollen folgende Eingabefelder des Backends ausgegeben werden:

- Frage

- Antwort

- Autor

- Beitragsbild (falls vorhanden)

Es werden alle FAQs auf einer Seite dargestellt. Die Anzahl der FAQs ist abhängig von der zugewiesenen Kategorie. Eine Pagination ist nicht gewünscht.

**>> Dokumentation**

Die Webapplikation soll intuitiv bedient werden können. Ein kleiner Benutzerleitfaden ist für spätere Kunden-Schulungszwecke trotzdem erforderlich. Für Entwickler soll eine technische Dokumentation erstellt werden.

**>> Testing**

Der Lernende erstellt im Zuge der IPA ein umfangreiches Testprotokoll, die Tests werden dabei selbstständig durchgeführt und dokumentiert. Auftretende Mängel werden vom Lernenden behoben. Der Lernende erstellt anhand der detaillierten Aufgabenbeschreibung die Anforderung des Testprotokolls.

## 2 Mittel und Methoden

### 2.1 Projektmethode *aus PkOrg*

Ich verwende die IPERKA Methode da sie sich sehr gut eignet, eine IPA durchzuführen. Sie legt sehr grossen Wert auf Planung und zudem hilft die Methode gut aus vorherigen Projekten zu lernen.

### 2.2 Zu verwendende Mittel und Methoden *aus PkOrg*

- WordPress 5.3.x

- Advanced Custom Field Plug-In

- PHP 7.3

- HTML

- CSS

- MySQL

- Programmiersoftware

- Internetdokumentationen

**2.2.1 Overall verwendete Tools** *aus PkOrg*

- WordPress CMS

- Standard WP Theme

- Impreza Theme

- GIT Console

- BrowserStack

- Browser

- Plesk

- Programmiersoftware

**2.2.2 Ergänzungen**

PHP Storm (IDE)

Mamp Pro (Lokale Entwicklungsumgebung)

## 3 Vorkenntnisse

### 3.1 Vorkenntnisse *aus PkOrg*

Kenntnisse in der unter «Mittel und Methoden» genannten Programmier- und Scriptsprachen, sowie vom Lernenden erworbene strukturelle Kenntnisse des WordPress CMS.

### 3.2 Neue Lerninhalte *aus PkOrg*

Keine

### 3.2 Vorarbeiten *aus PkOrg*

- Aufsetzen einer lokalen Entwicklungsumgebung inkl. Vorbereitung eines GIT

- Repositorys auf github zu Versionierungszwecken.

- Dokumentvorlagen (z.B. Protokoll) erstellen

**3.2.1 Weitere vorarbeiten**

Ich habe mir 5 IPA Dokumentationen angesehen, dabei habe ich mir Gedanken und Notizen gemacht, damit ich das Beste aus den 5 Dokumentationen mitnehmen kann.

### 3.3 Neue Lerninhalte *aus PkOrg*

Keine

### 3.4 Arbeiten in den letzten 6 Monaten *aus PkOrg*

- Aufbau von individuellen WordPress Themes und Plug-Ins mittels PHP, HTML und CSS

- Funktionelle und designspezifische Anpassungen an bestehenden WordPress Kundenwebseiten

- Aktivierung von SSL und daraus erfolgende Datenbankmutationen von WordPress Webseiten

- Ausgabeseitige Darstellung von Beitragsdaten in WordPress

**3.4.1 Dabei waren die grössten Arbeiten folgende**

>> Der vollständige Aufbau einer Kundenwebseite (https://daniel-ilg-malerei.ch/) in Rücksprache mit dem zuständigen Projektleiter und nach unseren Firmenkriterien. In diese Arbeit wurden ca. 25 Arbeitstage investiert.

>> Der vollständige und selbstständige Aufbau eines mehrsprachigen Mitarbeiter-Plug-Ins für WordPress mittels Shortcode Integration nach Firmenstandards. In diese Arbeit wurden ca. 15 Arbeitstage investiert.

### 3.6 Erfahrungen mit Wordpress

Während meiner Ausbildung konnte ich Komplexere Projekte wie das Programmieren und anpassen an bereits vorhandenen Themes und Plug-Ins bis zu einfachen Umstellungen an verschiedenen Websites bewältigen und vornehmen.

In meiner Freizeit habe ich auf meiner eigenen Seite an mehreren Projekten mit Wordpress gearbeitet, um meine Kenntnisse über das CMS zu stärken.

Jedoch fehlt es mir an Erfahrung die objektorientierte Programmierung einzusetzen, weshalb ich bisherige Projekte und Schulmodule analysieren werde.

## 4 Organisation

### 4.1 Onebyte AG

### 4.2 Projektablauforganisation

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Erstellt mit Omni Graffle

## 5 Zeitplanung

5.1 Zeitplan

5.2 Versionsverwaltung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IPA Tagesjournal**  Wordpress Plug-In «FAQ» | | | | |
| Tag 1 | | 27.04.2020 | | |
| **Ausgeführte Arbeiten** | | **Bemerkungen** | | |
| Projektübergabe | | Das Projekt wurde erfolgreich begonnen | | |
| Zeitplan erstellen | | Der Zeitplan wurde erstellt mit ersten Einträgen | | |
| Dokumentation erstellen | | Die Dokumentation wurde für Tag 1 abgeschlossen | | |
| Tages-Backup erstellt | | Das erste Backup wurde erstellt | | |
|  | |  | | |
|  | |  | | |
|  | |  | | |
| **Einhaltung des Zeitplans** | | | | |
| Die Einhaltung des heutigen Zeitplanes hat Reibungslos geklappt | | | | |
| **Hilfeleistungen** | | | | |
|  | | | | |
| **Benutztes Wissen & Hilfsbeschaffung** | | | | |
| Um etwas Zeit zu sparen habe ich als Vorlage einen ehemals Kopierten Plan genommen, welchen ich angepasst habe:  <https://github.com/AYIDouble/IPA-2018-Informatiker-EFZ-Applikationsentwicklung-Alpay-Yildirim/blob/master/IPA-2018-Zeitplan.pdf>  Um IPERKA im Blick zu halten hatte ich nebenbei folgende Seite offen:  <https://mischakraus.ch/eportfolio-modul-431/lern-planungstechniken/iperka/> | | | | |
| **Probleme** | | | | |
| Keine Probleme an diesem Tag | | | | |
| **Reflexion** | | | | |
| Ich hatte Mühe, den Zeitplan einzuschätzen. Ich habe mir dann frühere Projekte in Erinnerung gerufen welche ähnliche Teilabschnitte hatten und habe die Zeiten geplant, welche ich einschätzen konnte. Die restlichen Stunden habe ich durch eigene Einschätzung der Schwierigkeit auf die verbleibenden Teilabschnitte aufgeteilt. | | | | |
| **Arbeitsbeginn** | **Pausenbeginn** | | **Wiederaufnahme** | **Feierabend** |
| 09:00 | 12:00 | | 13:00 | 18:00 |

## 7 Tagesjournale