

Prüfungsphase: Sommer 2024

Name, Vorname: Gregory Maleterre
Ausbildungsberuf: Fachinformatiker für Anwendungsentwicklung

Prüflingsnummer: (183) 06455
Id.Nr.: 1586681-50

Effizientes E-Commerce: Erstellen, Erweitern und Realisieren einer Optimierten Produktverwaltung in WooCommerce mit Python, Excel und CSV

Projektgeber:

Pixelding
Provinzstraße 57

13409 Berlin

Ausbildungsbetrieb:

BFW Berlin-Brandenburg e. V.
Kastanienallee 25
16567 Mühlenbeck

Ansprechpartner:

Herr Poguntke
Telefon: +49 33056 86316

Mail: martin.poguntke@bawt-berlin-brandenburg.de

Eidesstattliche Erklärung:

Hiermit erkläre ich, Gregory Maleterre, die hier vorliegende Arbeit (Dokumentation) selbstständig und nur unter Zuhilfenahme der aufgeführten Quellen angefertigt zu haben. Der aufgeführte zeitliche Rahmen wurde eingehalten.

Berlin 17.04.2024

Gregory Maleterre

Ort/Datum

Unterschrift



Sommerprüfung 2024

Ausbildungsberuf

Fachinformatiker/Fachinformatikerin (VO 2020) Fachrichtung:
Anwendungsentwicklung

Prüfungsbezirk

Potsdam FI 1196-1 (AP T2V1)

Gregory Maleterre

Identnummer: 1586681-50

E-Mail: gmaleterre@yahoo.es, Telefon: +49 151 27141287

Ausbildungsbetrieb: Berufsförderungswerk Berlin Brandenburg e.V.

Projektbetreuer: Martin Pogunkte

E-Mail: martin.pogunkte@bfw-berlin-brandenburg.de, Telefon: +49 33056 86316

Thema der Projektarbeit

Effizientes E-Commerce: Erstellen, Erweitern und Realisieren einer optimierten
Produktverwaltung in WooCommerce mit Python, Excel und CSV.



Gregory Maletter
Identnummer: 1586681-50

16.04.2024

1 Thema der Projektarbeit

Effizientes E-Commerce: Erstellen, Erweitern und Realisieren einer optimierten Produktverwaltung in WooCommerce mit Python, Excel und CSV.

2 Geplanter Bearbeitungszeitraum

Beginn: 19.02.2024
Ende: 12.04.2024

3 Ausgangssituation

Ausgangssituation

Projektumfeld

Es ist genehmigt worden ist, dass diesen Projekt der IHK-Prüfungskommission vorgelegt werden darf.

Was macht die Firma

Projektgeber: Pixelding ist ein verlässlicher Partner in Sachen Web- und Grafikdesign aus der Medien-Hauptstadt Berlin, sie machen Webdesign, Online-Shops, Social Media, virtuelle Touren, Copyright Identity, Druckmedien, Fotografie, Werbekampagnen.

Wie groß ist die Firma

Der Boss, ein Programmierer, ein Grafiker, Praktikanten 4 – 5 für SEO und Grafik.

In welcher Abteilung wird das Projekt durchgeführt

Online-Shop, hauptsächlich Programmierung, Teil vom Projekt wird an SEO- und Grafikteam delegiert.

Mitarbeiter und Vorgesetzte, mit denen ich interagiere im Projekt

SEO-Team, Grafiker, Boss.

Kundenfirma

Die Kundenfirma ist im Verkauf tätig und will seine Onlinepräsenz verbessern, bzw. mehr verkaufen.

Personalplanung



Gregory Maletterre
Identnummer: 1586681-50

16.04.2024

Nur ich.

Projektbeschreibung

Das Projekt hat das Ziel, einen reibungslosen Prozess für die Integration neuer Produkte in einen bestehenden Online-Shop auf Basis von WordPress WooCommerce zu schaffen. Der Kunde plant, Produkte von neuen Lieferanten in seinen Shop aufzunehmen, wobei sämtliche Informationen zu den Produkten, einschließlich Name, SKU, Fotos, Booklet-PDFs, Preise und Beschreibungen erfasst werden sollen.

Der Workflow beginnt mit der automatischen Erstellung einer strukturierten Liste aller Produkte von den neuen Lieferanten. Diese Liste bildet die Grundlage für den weiteren Prozess. Fotos und PDF-Dateien werden in einem systematisch angelegten Ordner gespeichert, wodurch eine effiziente Organisation und Auffindbarkeit gewährleistet sind.

Parallel dazu werden zwei separate Excel-Dateien vorbereitet: eine für das SEO-Team, um Produktbeschreibungen zu aktualisieren, und eine für das Grafikdesign-Team, um bei Bedarf Bilder zu aktualisieren. Eine zusätzliche Excel-Datei wird speziell für den Kunden erstellt, in der er Preise, Versandkosten und Lagerbestände bequem eingeben kann.

Die entwickelte Lösung gewährleistet nicht nur, dass der gesamte Prozess effizient und benutzerfreundlich abläuft, sondern bietet auch die automatische Generierung der ersten Produktliste, um menschliche Eingriffe auf ein Minimum zu reduzieren. Auf diese Weise kann der Kunde mühelos neue Produkte in seinem Online-Shop präsentieren.

Ist-Zustand

Der gegenwärtige Zustand des Online-Shops basiert auf einem manuellen Prozess zur Erstellung der Produktliste. Dabei werden Namen und Informationen von den Websites der Großhändler kopiert und eingefügt. Sowohl der Download von PDFs als auch von Bildern erfolgt manuell und beansprucht erheblich viel Zeit. Die manuelle Benennung und Strukturierung der Ordner für die Speicherung von Bildern und PDFs führt zu potenziellen Tippfehlern und erschwert die Auffindbarkeit des richtigen Pfads erheblich.

Zusätzlich ist der Kunde gezwungen, seine Produkte einzeln in die WooCommerce-Plattform über die langsame grafische Benutzeroberfläche von WordPress einzugeben. Dieser mühsame Prozess erfordert nicht nur das manuelle Verknüpfen von Bildern, sondern auch das Kopieren und Einfügen von Preisen und anderen Informationen. Die resultierende Arbeitsweise ist äußerst zeitaufwendig und ineffizient.

Die manuelle Natur dieser Abläufe birgt das Risiko von menschlichen Fehlern, was zu ungenauen Produktinformationen und Verzögerungen bei der Aktualisierung der Produktliste führen kann. Insgesamt ist der aktuelle Prozess nicht nur zeitintensiv, sondern auch anfällig für Fehler und stellt eine erhebliche Herausforderung für eine effiziente Verwaltung und Aktualisierung des Produktbestands dar.

Personelle Schnittstellen



Gregory Maletter
Identnummer: 1586681-50

16.04.2024

Alex, der Boss, ist der Projektgeber.

Kostenplanung

Die Kostenplanung folgt in der Dokumentation.

Sachmittelplanung & Technische Schnittstellen

Ionus Server 10 Euro per Monat für die Webseite, Computer Windows Lizenz 150 Euro, WordPress ist umsonst, WooCommerce ist umsonst, Xampp ist umsonst. Python, GitHub, Visual Studio Code, Google Drive, Asana, Slack, FileZilla, phpMyAdmin sind auch umsonst.

4 Projektziel

Soll-Zustand

Unser Kunde benötigt tausende neue Produkte mit Fotos, PDF-Downloads und Preisen und Beschreibungen auf seiner Webseite. Er benötigt eine unkomplizierte Infrastruktur zur Preisgestaltung sowie die Möglichkeit, die Texte aller Produkte für SEO-Zwecke zu aktualisieren. Die optimierte Lösung automatisiert den Produktintegrationsprozess erheblich. Ein automatischer Algorithmus generiert eine strukturierte Liste aller neuen Produkte von den Lieferantenwebseiten. Sie bildet die Grundlage für sämtliche Aktualisierungen im System. Fotos und PDFs werden automatisch heruntergeladen und in einem systematisch organisierten Ordner abgelegt. Dadurch wird eine fehleranfällige manuelle Handhabung vermieden, und die Auffindbarkeit der Dateien wird optimiert. Zwei separate Exceldateien werden automatisch erstellt: Eine für das SEO-Team zur Aktualisierung von Produktbeschreibungen und eine für das Grafikdesign-Team zur optionalen Aktualisierung von Fotos. Die Exceldateien werden im Anschluss automatisch in das erforderliche CSV-Format für WooCommerce konvertiert. Dies schließt auch die spezielle Exceldatei des Kunden ein, um Preise, Versandkosten und Lagerbestände unkompliziert zu verwalten. Diese automatisierte Lösung gewährleistet nicht nur eine Zeitersparnis, sondern minimiert auch das Fehlerrisiko. Der gesamte Prozess ist effizienter und benutzerfreundlicher, sodass der Kunde mühelos neue Produkte in seinem Online-Shop präsentieren kann.

Projektaufgaben

Der Ablauf beinhaltet die Nutzung von Python zur Datenextraktion von den Webseiten der Lieferanten. Die gesammelten Daten werden in einer JSON-Datei gespeichert. Die Bilder und PDFs jedes Produkts werden in einem Ordner abgelegt, dessen Name sich aus dem Produktnamen ableitet, und der Pfad dazu wird in der JSON-Datei referenziert. JSON-File und Folder werden nicht nur auf Google Drive gesichert, sondern auch auf zwei separaten Festplatten für zusätzliche Redundanz abgelegt. Anschließend erfolgt die Umwandlung der JSON-Daten in das notwendige WooCommerce CSV-Format mittels Python. Die Dateien werden daraufhin auf den Test-WooCommerce-Shop im lokalen Host hochgeladen. Das Grafikteam erhält eine autogenerierte XLSX-Liste der Produktfotos zur eingehenden Prüfung. Gleichzeitig wird dem SEO-Team eine XLSX-Liste bereitgestellt, um Texte umzuschreiben und zu ergänzen. Parallel dazu wird eine Excel-Tabelle für den Kunden erstellt,



Gregory Maletterre
Identnummer: 1586681-50

16.04.2024

um die Verwaltung seiner Preise und Lagerbestände zu erleichtern. Schließlich wird ein System implementiert, das die Exceldateien wiederum in das WooCommerce CSV-Format konvertiert. Die CSV-Dateien werden lokal auf WordPress hochgeladen, um umfassende Tests durchzuführen. Bei erfolgreicher Durchführung werden die CSV-Dateien vom lokalen Shop exportiert und in den Online-Shop des Kunden übertragen. Nach erneuten Tests zur Sicherstellung eines reibungslosen Ablaufs werden sämtliche Projektdetails, Probleme und Fortschritte in Asana und Google Drive dokumentiert.

5 Zeitplanung

Zeitplanung

Technische Analyse des Ist-Zustandes: 5 Stunden

- Untersuchung des aktuellen manuellen Prozesses zur Integration neuer Produkte: 2 Stunden
- Evaluierung der bestehenden Datenextraktionsmethoden und -werkzeuge: 2 Stunden
- Meeting: 1 Stunde

Herstellen der Arbeitsumgebung: 3 Stunden

- Aufsetzen einer geeigneten Entwicklungsumgebung, einschließlich der erforderlichen Softwaretools wie Python, BeautifulSoup oder Selenium, Xampp: 3 Stunden

Datenextraktion von 4 Webseiten: 31 Stunden

- Python Script schreiben: 10 Stunden
- Durchführung von Modultests auf die Datenextraktion mittels Python: 5 Stunden
- Durchführung eines Systemtests, um die gesamte Datenextraktionspipeline zu überprüfen: 5 Stunden
- Script anpassen und herunterladen: 9 Stunden
- Sicherung auf Google Drive sowie auf zwei separaten Festplatten für zusätzliche Redundanz: 1 Stunde
- Meeting: 1 Stunde

Datenkonvertierung für WooCommerce: 11 Stunden

- Umwandlung der JSON-Daten in das erforderliche WooCommerce CSV-Format mittels Python: 7 Stunden
- WordPress Localhost-Shop installieren: 2 Stunden
- Tests: 1 Stunde
- Hochladen der Dateien auf den Test-WooCommerce-Shop im lokalen Host: 1 Stunde

XLSX-Liste der Produktfotos für das Grafikerteam: 1 Stunde

- Bereitstellung einer XLSX-Liste der Produktfotos zur eingehenden Prüfung: 1 Stunde



Gregory Maletter
Identnummer: 1586681-50

16.04.2024

XLSX-Liste der Texte für das SEO-Team: 1 Stunde

- Bereitstellung einer XLSX-Liste für das SEO-Team zur Überarbeitung und Ergänzung von Texten: 1 Stunde

Erstellung einer Excel-Tabelle für den Kunden: 1 Stunde

- Erstellung einer Excel-Tabelle zur erleichterten Verwaltung von Preisen und Lagerbeständen durch den Kunden: 1 Stunde

Implementierung eines Systems für die Datenrückkonvertierung: 2 Stunden

- Implementierung eines Systems zur Konvertierung der bearbeiteten Excel-Dateien zurück in das WooCommerce CSV-Format: 2 Stunden

Lokale Tests auf WordPress: 7 Stunden

- Hochladen der CSV-Dateien lokal auf WordPress: 1 Stunde
- umfassende Tests: 6 Stunden

Export und Übertragung in den Online-Shop des Kunden: 2 Stunden

- Export der CSV-Dateien vom lokalen Shop und Übertragung in den Online-Shop des Kunden: 2 Stunden

Erneute Tests zur Sicherstellung eines reibungslosen Ablaufs: 6 Stunden

- Durchführung erneuter Tests, um die ordnungsgemäße Funktion sicherzustellen: 6 Stunden

Dokumentation von Projektdetails, Problemen und Fortschritten: 10 Stunden

- Dokumentation sämtlicher Projektdetails, auftretender Probleme und erzielter Fortschritte in Asana und Google Drive. 9 Stunden
- Meeting: 1 Stunde

Gesamt: 80 Stunden

6 Anlagen

siehe Anlage 1

7 Präsentationsmittel

Laptop



Gregory Maletter
Identnummer: 1586681-50

16.04.2024

8 Hinweis!

Ich bestätige, dass der Projektantrag dem Ausbildungsbetrieb vorgelegt und vom Ausbildenden genehmigt wurde. Der Projektantrag enthält keine Betriebsgeheimnisse. Soweit diese für die Antragstellung notwendig sind, wurden nach Rücksprache mit dem Ausbildenden die entsprechenden Stellen unkenntlich gemacht.

Mit dem Absenden des Projektantrages bestätige ich weiterhin, dass der Antrag eigenständig von mir angefertigt wurde. Ferner sichere ich zu, dass im Projektantrag personenbezogene Daten (d. h. Daten über die eine Person identifizierbar oder bestimmbar ist) nur verwendet werden, wenn die betroffene Person hierin eingewilligt hat.

Bei meiner ersten Anmeldung im Online-Portal wurde ich darauf hingewiesen, dass meine Arbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit „null“ Punkten bewertet werden kann. Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Arbeit im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Es ist mir bewusst, dass Kontrollen durchgeführt werden.

9 Grund für „mit Auflage genehmigt“

Guten Tag Gregory Maletter,
gehen Sie in Ihrer Dokumentation bitte genauer auf den Punkt "Datenextraktion von den Webseiten der Lieferanten" ein. Warum wurde sich für diesen Weg entschieden und warum nicht für eine definiertere Kommunikation über z.B. Schnittstellen?



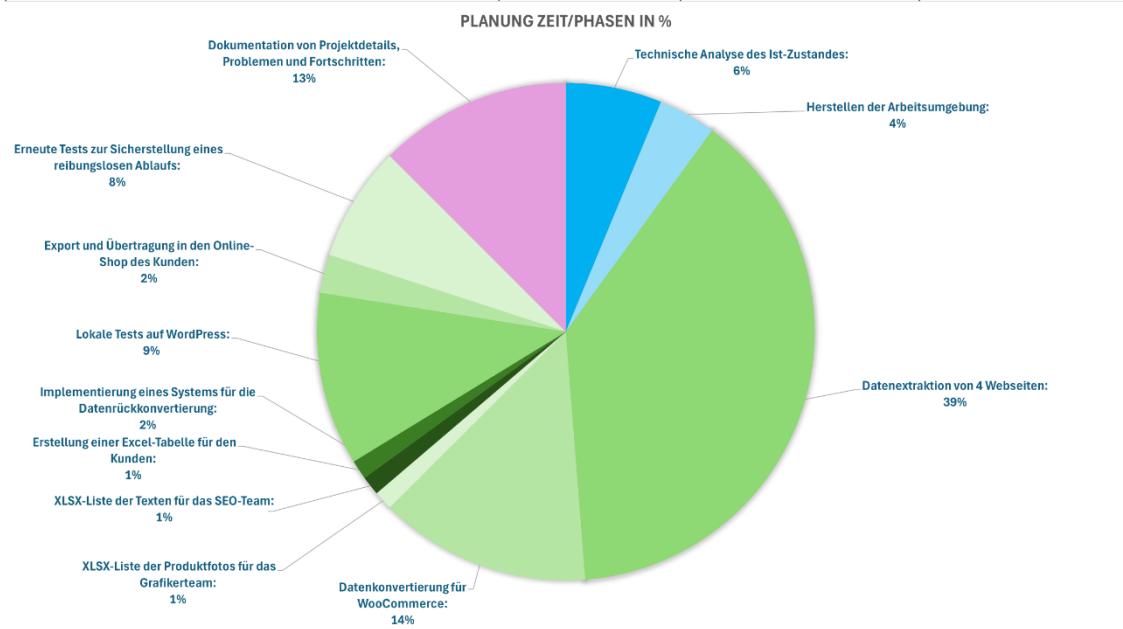
Herr Gregory Maleterre
Identnummer: 1586681-50

16.04.2024

Anlage 1

Grafische und tabellarische Darstellung

Phase	Dauer in Stunden	Phase	Dauer in Stunden
Technische Analyse des Ist-Zustandes:	5		
Herstellen der Arbeitsumgebung:	3	Planungsphase	8
Datenextraktion von 4 Webseiten:	31	Umsetzungsphase	62
Datenkonvertierung für WooCommerce:	11		
XLSX-Liste der Produktfotos für das Grafikteam:	1		
XLSX-Liste der Texten für das SEO-Team:	1		
Erstellung einer Excel-Tabelle für den Kunden:	1		
Implementierung eines Systems für die Datenrückkonvertierung:	2		
Lokale Tests auf WordPress:	7		
Export und Übertragung in den Online-Shop des Kunden:	2		
Erneute Tests zur Sicherstellung eines reibungslosen Ablaufs:	6		
Dokumentation von Projektdetails, Problemen und Fortschritten:	10	Abgabe/ Dokumentationsphase	10
Summe	80	Summe	80



Zeitmitschreibung der Projektarbeit

Prüfungsteilnehmer

Name: Gregory

Vorname: Maleterre

Login: 1831586681-50

Datum	Tätigkeit	Zeit in Stunden
	Planungsphase	
Mo. 19 Februar	Technische Analyse des Ist-Zustandes	5
Mo. 19 Februar	Herstellen der Arbeitsumgebung	2
Die. 20 Februar	Herstellen der Arbeitsumgebung	1
	Summe Planungsphase	8
	Durchführungsphase	
Die. 20 Februar	Datenextraktion von der 1. Webseite	6
Mit. 21 Februar	Datenextraktion von der 1. Webseite	2
Mit. 21 Februar	Datenextraktion von der 2. Webseite	5
Don. 22 Februar	Datenextraktion von der 2. Webseite	2
Don. 22 Februar	Datenextraktion von der 3. Webseite	5
Fre. 23 Februar	Datenextraktion von der 3. Webseite	5
Fre. 23 Februar	Datenextraktion von der 4. Webseite	2

Mon. 26 Februar	Datenextraktion von der 4. Webseite	5
Die 27 Februar	Datenkonvertierung für WooCommerce	7
Mit 28 Februar	Datenkonvertierung für WooCommerce	4
Don 29 Februar	XLSX-Liste der Produktfotos für das Grafikerteam	0.75
Don 29 Februar	XLSX-Liste der Texte für das SEO-Team	0.75
Don 29 Februar	Erstellung einer Excel-Tabelle für den Kunden	1
Mon. 18 März	Implementierung eines Systems für die Datenrückkonvertierung	2
Mon. 18 März	Lokale Tests auf WordPress	5
Die 19 März	Lokale Tests auf WordPress	2
Die 19 März	Export und Übertragung in den Online-Shop des Kunden	2
Die 19 März	Erneute Tests zur Sicherstellung eines reibungslosen Ablaufs	3
Mit 20 März	Erneute Tests zur Sicherstellung eines reibungslosen Ablaufs	3
	Summe Durchführungsphase	62.5
	Übergabe-, Auswertungsphase	
Mit 20 März	Dokumentation von Projektdetails, Problemen und Fortschritten	4
Don 21 März	Dokumentation von Projektdetails, Problemen und Fortschritten	6
	Summe Übergabe-, Auswertungsphase	10
	Summe gesamt	80.5

Inhaltsverzeichnis

1. Projektdokumentationshinweise Verzeichnisse	1
1.1. Verwendungsvermerk	1
1.2. Dokumentation zur Projektarbeit	1
2. Planungsphase	1
2.1. Ausgangssituation.....	1
2.1.1. Projektumfeld.....	1
2.1.2. IST-Zustand	2
2.1.3. SOLL-Zustand	2
2.1.4. Projektziele	3
2.1.5. Personellen Schnittstellen	3
2.1.6. Technische Schnittstellen	3
2.1.7. Projektabgrenzung	4
2.1.8. Projektbegründung	5
2.2. Projektablauf	5
2.2.1. Technische Planung des Projektziels	5
3. Ressourcen- und Ablaufplan.....	6
3.1. Personalplanung	6
3.2. Sachmittelplanung.....	6
3.3. Zeitliche Planung.....	6
3.4. Kostenplanung	7
4. Durchführungsphase	8
4.1. Herstellen der Arbeitsumgebung	8
4.2. Eigentlich Umsetzung	8
4.2.1. Daten von den Webseiten holen	8
4.2.1.1 Navigation auf der Webseite mit Python.....	9
4.2.1.2 Erstellen von Ordnern für Produkte, PDFs und Bilder.....	9
4.2.1.3 Download.....	9
4.2.2. Daten in lokale WordPress importieren.....	10
4.2.2.1 lokale WordPress installieren.....	10
4.2.2.2 Daten für WordPress WooCommerce in CSV, und XLSX umbauen	10
4.2.2.3 CSV in WordPress WooCommerce importieren.....	11
4.2.3. XLSX für Grafiker, SEO, und Kunde anfertigen	11
4.2.3.1 XLSX-Liste für den Kunden.....	12
4.2.3.2 XLSX-Liste für das SEO-Team	12
4.2.3.3 XLSX-Liste für die Grafiker	12
4.2.4. Daten in lokale WordPress updaten.....	12
4.2.4.1 Erstellen einer neuen lokale WordPress-WooCommerce-Instanz	12

4.2.4.2 Konvertierung der XLSX-files zu CSV-files mit Python	12
4.2.4.3 Hochladen der CSV in den neuen Lokalen WordPress.....	12
4.2.5. Daten in Onlineshop hochladen	12
4.2.5.1 Exportieren der CSV-File aus dem Lokal Shop	13
4.2.5.2 Anpassen der Bilder und PDF URL in der CSV-File	13
4.2.5.3 Onlinelagerung der Bilder und PDF	13
4.2.5.4 Import der CSV-File in den Onlineshop	13
4.3. Testen	13
5. Übergabe und Auswertung.....	14
5.1. Projektübergabe.....	14
5.2. Mitarbeiter Schulung, Bedienungsanleitung.....	14
5.3. Soll-Ist-Vergleich (inhaltlich, Zeitlich, kostentechnisch).....	14
5.4. Technische Auswertung und Ausblick.....	15
5.5. Persönliche Auswertung	15

Abbildungsverzeichnis

Abbildung 1: Aktivität Diagramm	Error! Bookmark not defined.
Abbildung 2: Programmierer Stundenlohn.....	B
Abbildung 3: Head of IT-Stundenlohn	B
Abbildung 4: Home-office-Pauschale.....	C
Abbildung 5: Aktivität Diagramm	D
Abbildung 6: URL-Programmierung	E
Abbildung 7: Log Output.....	F
Abbildung 8: Erstellen von Ordnern für Produkte, PDFs und Bilder Programmierung.....	G
Abbildung 9: Folders Anzahl	G
Abbildung 10: Download Programmierung	Error! Bookmark not defined.
Abbildung 11: JSON-Output.....	Error! Bookmark not defined.
Abbildung 12: Protokoll	Error! Bookmark not defined.
Abbildung 13: JSON	Error! Bookmark not defined.
Abbildung 14: Protokoll	Error! Bookmark not defined.
Abbildung 15: JSON-Output.....	Error! Bookmark not defined.
Abbildung 16: Protokoll	L
Abbildung 17: Produkt CSV Import Schema.....	M
Abbildung 18: Aktivität Diagramm	R
Abbildung 19: CSV Output	S
Abbildung 20: WooCommerce Import GUI 1	S
Abbildung 21: WooCommerce Import GUI 2	T
Abbildung 22: WooCommerce Import GUI 3	T
Abbildung 23: XLSX zu CSV	U

Abkürzungen

CSV - Comma-separated Values

FTP - File Transfer Protocol

GUI - Graphical User Interface

HTML - Hypertext Markup Language

JSON - JavaScript Object Notation

PDF - Portable Document Format

SEO - Search Engine Optimization

SKU - Stock Keeping Unit

URL - Uniform Ressource Locator

UTF-8 - 8-bit Unicode Transformation Format

XAMPP - Cross-Plattform (X), Apache (A), MySQL (M), PHP (P), and Perl (P)

XLSX - Microsoft Excel Open XML Spreadsheet

Quellenverzeichnis

<https://www.stepstone.de/gehalt/Programmierer-in.html>

<https://www.stepstone.de/gehalt/Head-of-IT.html>

<https://www.test.de/Homeoffice-Pauschale-So-rechnen-Sie-mit-dem-Finanzamt-ab-5788023-0/>

<https://github.com/woocommerce/woocommerce/wiki/Product-CSV-Import-Schema#csv-columns-and-formatting>

Effizientes E-Commerce: Erstellen, Erweitern & Realisieren einer Optimierten Produktverwaltung in WooCommerce mit Python, Excel und CSV

19. Februar - 12. April 2024

Gregory Maleterre

Vorwort/ Ihre Frage

Gehen Sie in Ihrer Dokumentation bitte genauer auf den Punkt "Datenextraktion von den Webseiten der Lieferanten" ein. Warum wurde sich für diesen Weg entschieden und warum nicht für eine definierter Kommunikation über z.B. Schnittstellen?

Es gibt einen klaren Unterschied zwischen den Webseiten der Hersteller und denen der Lieferanten. Während es bei den Herstellern oft die Möglichkeit einer definierten Kommunikation über Schnittstellen gibt, ist dies bei den Lieferanten nicht der Fall. Diejenigen, die diese Option anbieten, stellen in der Regel nur begrenzte Informationen zur Verfügung, die eher für ein Fachpublikum gedacht sind.

Das SEO-Team ist möglicherweise nicht mit dem Spezialgebiet der Putzmittel vertraut und benötigt daher umfassendere Informationen, die es auch für ein breiteres Publikum verständlich aufbereiten kann. In Anbetracht dieser Anforderungen erwies sich die Extraktion von Informationen und Daten mit Python als die bessere Lösung.

Die Verwendung von Python ermöglichte eine flexible und anpassungsfähige Extraktion direkt von den Webseiten der Lieferanten. Auf diese Weise konnten wir sicherstellen, dass das SEO-Team über die notwendigen Informationen verfügte, um die Produktbeschreibungen optimal zu gestalten und gleichzeitig die Anforderungen des Kunden zu erfüllen. Diese Herangehensweise bot uns die Möglichkeit, eine umfassende Datengrundlage zu schaffen, die sowohl für interne als auch externe Zwecke geeignet war.

1. Projektdokumentationshinweise Verzeichnisse

1.1. Verwendungsvermerk

Es wurde in Absprache mit dem Projektgeber, Pixelding, und dem Kunden, Putzmittelshop24, eine Vereinbarung getroffen und genehmigt, dass ich ihre Daten für die Dokumentation meines Projekts verwenden darf.

1.2. Dokumentation zur Projektarbeit

Das Projekt zielt darauf ab, einen reibungslosen Prozess für die Integration neuer Produkte in einen bestehenden Online-Shop auf Basis von WordPress WooCommerce zu schaffen. Eine prozessorientierte Dokumentation wird erstellt. Ich werde die einzelnen Schritte nach dem Wasserfallmodell dokumentieren.

2. Planungsphase

2.1. Ausgangssituation

2.1.1. Projektumfeld

Pixelding ist ein verlässlicher Partner in Sachen Web- und Grafikdesign aus der Medien-Hauptstadt Berlin. Mit einem Team von 2–10 Beschäftigten vereinen sie Expertise, Leidenschaft und effektive Kommunikation, um die Visionen ihrer Kunden zum Leben zu erwecken. Ihre Kernkompetenzen umfassen Webdesign, Suchmaschinenoptimierung (SEO), die Entwicklung virtueller Touren sowie die Gestaltung und Umsetzung von Online-Shops und Buchungssystemen.

Zusätzlich bieten sie professionelle Werbekampagnen und unterstützen bei der Gestaltung von Printmedien. Weitere Informationen über PixelDing finden Sie auf unserer Website unter:
<https://www.pixelding.de>

Das Projekt umfasst die Erweiterung eines Online-Shops und wird in verschiedenen Abteilungen verarbeitet. Alexander Wolobuew, der Eigentümer von PixelDing, ist der verantwortliche Leiter des Projekts, welches größtenteils von der Abteilung für Programmierung durchgeführt wird. Die Anpassung von Bildern liegt in der Zuständigkeit des Grafikteams, während das SEO-Team für die Textanpassungen verantwortlich ist.

Putzmittelshop24 ist ein etablierter Online-Shop für Reinigungsprodukte mit Sitz in Berlin. Der Shop wird von Thomas Pries geleitet und bietet eine breite Palette hochwertiger Reinigungsmittel für private Haushalte und gewerbliche Kunden.

Weitere Informationen über Putzmittelshop24 finden Sie auf unserer Website unter: <https://putzmittelshop24.de/>

2.1.2. IST-Zustand

Der gegenwärtige Zustand des Online-Shops basiert auf einem manuellen Prozess zur Erstellung der Produktliste. Dabei werden Namen und Informationen von den Websites der Großhändler kopiert und eingefügt. Sowohl der Download von PDFs als auch von Bildern erfolgt manuell und beansprucht erheblich viel Zeit. Die manuelle Benennung und Strukturierung der Ordner für die Speicherung von Bildern und PDFs führt zu potenziellen Tippfehlern und erschwert die Auffindbarkeit des richtigen Pfads erheblich.

Zusätzlich ist der Kunde gezwungen, seine Produkte einzeln in die WooCommerce-Plattform über die langsame grafische Benutzeroberfläche von WordPress einzugeben. Dieser mühsame Prozess erfordert nicht nur das manuelle Verknüpfen von Bildern, sondern auch das Kopieren und Einfügen von Preisen und anderen Informationen. Die resultierende Arbeitsweise ist äußerst zeitaufwendig und ineffizient.

Die manuelle Natur dieser Abläufe birgt das Risiko von menschlichen Fehlern, was zu ungenaunen Produktinformationen und Verzögerungen bei der Aktualisierung der Produktliste führen kann. Insgesamt ist der aktuelle Prozess nicht nur zeitintensiv, sondern auch anfällig für Fehler und stellt eine erhebliche Herausforderung für eine effiziente Verwaltung und Aktualisierung des Produktbestands dar.

2.1.3. SOLL-Zustand

Unser Kunde benötigt sämtliche tausende neue Produkte mit Fotos, PDF-Downloads und Preisen und Beschreibungen auf seiner Webseite. Darüber hinaus benötigt er eine unkomplizierte Infrastruktur zur Preisgestaltung sowie die Möglichkeit, die Texte aller Produkte für SEO-Zwecke zu aktualisieren.

Die optimierte Lösung automatisiert den Prozess der Produktintegration in den Online-Shop erheblich. Ein automatischer Algorithmus generiert eine strukturierte Liste aller neuen Produkte von den Lieferantenwebseiten. Diese Liste bildet die Grundlage für sämtliche Aktualisierungen im System.

Fotos und PDFs werden automatisch heruntergeladen und in einem systematisch organisierten Ordner abgelegt. Dadurch wird eine fehleranfällige manuelle Handhabung vermieden, und die Auffindbarkeit der Dateien wird optimiert.

Zwei separate Excel-Dateien werden automatisch erstellt: Eine für das SEO-Team zur Aktualisierung von Produktbeschreibungen und eine für das Grafikdesign-Team zur optionalen Aktualisierung von Bildern. Die Excel-Dateien werden im Anschluss automatisch in das erforderliche CSV-Format für WooCommerce konvertiert.

Dies schließt auch die spezielle Excel-Datei des Kunden ein, um Preise, Versandkosten und Lagerbestände unkompliziert zu verwalten. Diese automatisierte Lösung gewährleistet nicht nur eine erhebliche Zeitsparnis, sondern minimiert auch das Risiko von Fehlern.

Der gesamte Prozess ist effizienter und benutzerfreundlicher, sodass der Kunde mühelos neue Produkte in seinem Online-Shop präsentieren kann.

2.1.4. Projektziele

Aufgrund der fehlenden Option einer definierten Kommunikation über beispielsweise Schnittstellen, beinhaltet der Prozess die Verwendung von Python zur Extraktion von Daten von den Webseiten der Lieferanten. Die gesammelten Daten werden in einer JSON-Datei gespeichert. Die Bilder und PDFs jedes Produkts werden in einem Ordner abgelegt, dessen Name sich aus dem Produktnamen ableitet, und der Pfad dazu wird in der JSON-Datei referenziert. JSON-File und Folder werden nicht nur auf Google Drive gesichert, sondern auch auf zwei separaten Festplatten für zusätzliche Redundanz abgelegt. Anschließend erfolgt die Umwandlung der JSON-Daten in das notwendige WooCommerce CSV-Format mittels Python. Die Dateien werden daraufhin auf den Test-WooCommerce-Shop im lokalen Host hochgeladen.

Das Grafiker Team erhält eine autogenerierte XLSX-Liste der Produktfotos zur eingehenden Prüfung. Gleichzeitig wird dem SEO-Team eine XLSX-Liste bereitgestellt, um Texte umzuschreiben und zu ergänzen. Parallel dazu wird eine Excel-Tabelle für den Kunden erstellt, um die Verwaltung seiner Preise und Lagerbestände zu erleichtern.

Schließlich wird ein System implementiert, das die Excel-Dateien wiederum in das WooCommerce CSV-Format konvertiert. Die CSV-Dateien werden lokal auf WordPress hochgeladen, um umfassende Tests durchzuführen. Bei erfolgreicher Durchführung werden die CSV-Dateien vom lokalen Shop exportiert und in den Online-Shop des Kunden übertragen. Nach erneuten Tests zur Sicherstellung eines reibungslosen Ablaufs werden sämtliche Projektdetails, Probleme und Fortschritte in Asana und Google Drive dokumentiert.

2.1.5. Personellen Schnittstellen

In diesem Projekt habe ich unter der Verantwortung von Alexander Wolobuew gearbeitet, parallel dazu auch mit den Grafikern und SEO-Spezialisten.

2.1.6. Technische Schnittstellen

- **Python:** Eine Programmiersprache, die häufig für die Automatisierung, Datenanalyse und Webentwicklung verwendet wird.
- **WordPress:** Eine Open-Source-Plattform zur Erstellung und Verwaltung von Websites und Blogs.

- **WooCommerce:** Ein E-Commerce-Plugin für WordPress, das es ermöglicht, einen Online-Shop zu erstellen und zu betreiben.
- **IONOS Server:** Ein Open Network Operating System, das zur Steuerung und Verwaltung von Netzwerken verwendet wird.
- **Windows:** Ein Betriebssystem von Microsoft, das auf PCs weit verbreitet ist.
- **XAMPP:** Eine kostenlose und Open-Source-Software, die einen lokalen Server für die Entwicklung von Webanwendungen bereitstellt.
- **GitHub:** Eine Plattform für die Versionsverwaltung und Zusammenarbeit bei der Softwareentwicklung.
- **Visual Studio Code:** Ein plattformübergreifender Code-Editor, der von Entwicklern für die Softwareentwicklung verwendet wird.
- **Google Drive:** Ein cloudbasierter Speicherdiens von Google, der die Speicherung und gemeinsame Nutzung von Dateien ermöglicht.
- **Asana:** Eine webbasierte Anwendung zur Teamkollaboration und Aufgabenverwaltung.
- **Slack:** Ein Messaging-Tool für Teams, das die Kommunikation und Zusammenarbeit unterstützt.
- **FileZilla:** Eine kostenlose und Open-Source-FTP-Software zur Dateübertragung zwischen Computern und Servern.
- **phpMyAdmin:** Ein webbasiertes Administrationswerkzeug für die Verwaltung von MySQL-Datenbanken.

2.1.7. Projektabgrenzung

Dieses Projekt ist Teil eines größeren Vorhabens, das darauf abzielt, den Online-Shop von Putzmittel24 zu erweitern und verschiedene zusätzliche Funktionen und Inhalte bereitzustellen. Innerhalb dieses größeren Projektrahmens sind bestimmte Grenzen und Abgrenzungen definiert, um den Fokus und die Ziele klar zu halten. Zu den wichtigsten Aspekten der Projektabgrenzung gehören:

Erweiterung des Online-Shops: Das Hauptziel des Projekts ist die Integration neuer Produkte in den bestehenden Online-Shop von Putzmittel24. Dabei sollen sämtliche Informationen zu den Produkten, einschließlich Namen, SKU, Fotos, Booklet-PDFs und Preise, erfasst werden.

Bereitstellung von Reinigungstipps: Putzmittel24 plant die Einrichtung eines Informationsblocks auf seiner Website, der Reinigungstipps enthält. Dieser Block soll den Kunden zusätzliche nützliche Informationen bieten und das Engagement auf der Website erhöhen.

Video-Kanal auf YouTube: Um die Reichweite zu erhöhen und Kunden anzusprechen, wird auch die Einrichtung eines Video-Kanals auf YouTube geplant. Hier sollen Videos mit Anwendungstipps, Produktvorstellungen und anderen relevanten Inhalten veröffentlicht werden.

Präsenz in sozialen Medien: Zusätzlich zur YouTube-Plattform strebt Putzmittel24 eine Präsenz auf Facebook und Instagram an. Durch regelmäßige Beiträge und Interaktionen mit Kunden soll die Bekanntheit der Marke gesteigert und die Kundenbindung verbessert werden.

Verbesserung der Website für Barrierefreiheit: Ein weiteres wichtiges Ziel des Projekts ist die Verbesserung der Website von Putzmittel24 im Hinblick auf Barrierefreiheit. Dies umfasst die Anpassung der Website, um sie für Menschen mit Behinderungen zugänglicher zu machen und sicherzustellen, dass sie die Website ohne Einschränkungen nutzen können.

Ich bin nur für die Erweiterung des Online-Shops zuständig.

2.1.8. Projektbegründung

Die Projektbegründung basiert auf einem klaren Ist-Zustand, der die ineffizienten und fehleranfälligen manuellen Prozesse bei der Verwaltung des Produktbestands des Online-Shops verdeutlicht. Der gegenwärtige Zustand zeigt, dass die manuelle Extraktion von Produktinformationen von den Websites der Großhändler zeitaufwendig ist und ein hohes Risiko für menschliche Fehler birgt. Dies führt zu ungenauen Produktinformationen und Verzögerungen bei der Aktualisierung der Produktliste.

Im Soll-Zustand wird deutlich, dass der Kunde eine umfassende Lösung benötigt, um den Prozess der Produktintegration erheblich zu optimieren und zu automatisieren. Die Anforderungen umfassen die automatisierte Extraktion von Produktinformationen von den Lieferantenwebseiten, das Herunterladen und Speichern von Fotos und PDFs, die automatische Generierung von Excel-Dateien zur Aktualisierung von Produktbeschreibungen und Bildern sowie die Konvertierung in das erforderliche CSV-Format für WooCommerce. Diese Lösung bietet nicht nur eine erhebliche Zeitersparnis, sondern minimiert auch das Risiko von Fehlern und verbessert die Effizienz der Produktverwaltung erheblich.

Die Entscheidung, Python für die Implementierung dieser Lösung zu verwenden, ergibt sich aus der Tatsache, dass eine definierte Kommunikation über Schnittstellen nicht möglich ist, da keine Zugangsdaten vorhanden sind. Somit ist Python die geeignete Plattform, um die automatisierten Prozesse effektiv umzusetzen und die Anforderungen des Kunden zu erfüllen.

2.2. Projektablauf

2.2.1. Technische Planung des Projektziels

Die technische Planung des Projektziels beinhaltet die detaillierte Analyse und Planung der Schritte, die erforderlich sind, um das definierte Ziel des Projekts zu erreichen. Dies umfasst die Auswahl der geeigneten Technologien, Tools und Methoden sowie die Festlegung des Arbeitsablaufs und der Ressourcen.

Im vorliegenden Projektziel, die Automatisierung des Produktintegrationsprozesses in den Online-Shop des Kunden, stehen folgende technische Schritte im Vordergrund:

1. **Auswahl der Programmiersprache und Frameworks:** Die Wahl fällt auf Python als Hauptprogrammiersprache aufgrund seiner Vielseitigkeit und der verfügbaren Bibliotheken für Web-Scraping und Datenverarbeitung. Als Framework wird BeautifulSoup für das Web-Scraping und pandas für die Datenverarbeitung verwendet.
2. **Implementierung des Web-Scraping-Algorithmus:** Entwicklung eines Python-Programms, das automatisch die Webseiten der Lieferanten durchsucht, die benötigten Produktinformationen extrahiert und in einer strukturierten Form speichert.
3. **Automatisierte Dateiverwaltung:** Einrichtung eines Systems zur automatischen Speicherung von heruntergeladenen Fotos und PDFs in einem ordnungsgemäß organisierten Ordnerstrukturen, um die Auffindbarkeit und Organisation zu optimieren.
4. **Entwicklung von Excel-Exportfunktionen:** Implementierung von Funktionen zur automatischen Generierung von Excel-Dateien für das SEO-Team und das Grafikdesign-Team, die für die Aktualisierung von Produktbeschreibungen und Bildern zuständig sind.

5. **Integration mit WooCommerce:** Konvertierung der Excel-Dateien in das erforderliche CSV-Format für WooCommerce und automatische Aktualisierung des Produktbestands im Online-Shop des Kunden.
6. **Testen und Validierung:** Durchführung von umfassenden Tests, um sicherzustellen, dass der entwickelte Algorithmus fehlerfrei funktioniert und den Anforderungen des Kunden entspricht. Validierung des Systems durch den Kunden, um sicherzustellen, dass es seinen Anforderungen entspricht.

3. Ressourcen- und Ablaufplan

3.1. Personalplanung

An der Programmierung der Python CSV- und XLSX-Funktionen arbeite ich alleine.

3.2. Sachmittelplanung

In der Sachmittelplanung werden die benötigten Ressourcen für das Projekt identifiziert, beschafft und verwaltet. Dazu gehören physische Gegenstände wie Computer, für die bereits vorhandene Geräte verwendet werden, sowie Softwarelizenzen wie WooCommerce und Visual Studio Code, die bereits bezahlt wurden oder kostenlos verfügbar sind.

3.3. Zeitliche Planung

Das Projekt ist geplant, vom 19. Februar 2024 bis zum 12. April 2024 abgeschlossen zu werden.

Zeitplanung

Technische Analyse des Ist-Zustandes: 5 Stunden

- Untersuchung des aktuellen manuellen Prozesses zur Integration neuer Produkte: 2 Stunden
- Evaluierung der bestehenden Datenextraktionsmethoden und -werkzeuge: 2 Stunden
- Meeting: 1 Stunde

Herstellen der Arbeitsumgebung: 3 Stunden

- Aufsetzen einer geeigneten Entwicklungsumgebung, einschließlich der erforderlichen Software-tools wie Python, BeautifulSoup oder Selenium, Xampp: 3 Stunden

Datenextraktion von 4 Webseiten: 31 Stunden

- Python Skript schreiben: 10 Stunden
- Durchführung von Modultests auf die Datenextraktion mittels Python: 5 Stunden
- Durchführung eines Systemtests, um die gesamte Datenextraktionspipeline zu überprüfen: 5 Stunden
- Skript anpassen und herunterladen: 9 Stunden
- Sicherung auf Google Drive sowie auf zwei separaten Festplatten für zusätzliche Redundanz: 1 Stunde
- Meeting: 1 Stunde

Datenkonvertierung für WooCommerce: 11 Stunden

- Umwandlung der JSON-Daten in das erforderliche WooCommerce CSV-Format mittels Python: 7 Stunden

- WordPress Localhost-Shop installieren: 2 Stunden
- Tests: 1 Stunde
- Hochladen der Dateien auf den Test-WooCommerce-Shop im lokalen Host: 1 Stunde

XLSX-Liste der Produktfotos für das Grafikerteam: 1 Stunde

- Bereitstellung einer XLSX-Liste der Produktfotos zur eingehenden Prüfung: 1 Stunde

XLSX-Liste der Texte für das SEO-Team: 1 Stunde

- Bereitstellung einer XLSX-Liste für das SEO-Team zur Überarbeitung und Ergänzung von Texten: 1 Stunde

Erstellung einer Excel-Tabelle für den Kunden: 1 Stunde

- Erstellung einer Excel-Tabelle zur erleichterten Verwaltung von Preisen und Lagerbeständen durch den Kunden: 1 Stunde

Implementierung eines Systems für die Datenrückkonvertierung: 2 Stunden

- Implementierung eines Systems zur Konvertierung der bearbeiteten Excel-Dateien zurück in das WooCommerce CSV-Format: 2 Stunden

Lokale Tests auf WordPress: 7 Stunden

- Hochladen der CSV-Dateien lokal auf WordPress: 1 Stunde
- umfassende Tests: 6 Stunden

Export und Übertragung in den Online-Shop des Kunden: 2 Stunden

- Export der CSV-Dateien vom lokalen Shop und Übertragung in den Online-Shop des Kunden: 2 Stunden

Erneute Tests zur Sicherstellung eines reibungslosen Ablaufs: 6 Stunden

- Durchführung erneuter Tests, um die ordnungsgemäße Funktion sicherzustellen: 6 Stunden

Dokumentation von Projektdetails, Problemen und Fortschritten: 10 Stunden

- Dokumentation sämtlicher Projektdetails, auftretender Probleme und erzielter Fortschritte in Asana und Google Drive. 9 Stunden
- Meeting: 1 Stunde

Gesamt: 80 Stunden

Abbildung 1: Aktivität Diagramm

3.4. Kostenplanung

Person	Stundenlohn	Anzahl Stunden	Total
ich	24,69 €	80	1.975,2 €
Chef	42,5 €	3	127,5 €
Sub total			2.102,7 €
Energie pauschal pro Tag 80/8 * 6			60 €
TOTAL			2.162,7 €

Quelle: <https://www.stepstone.de/gehalt/Programmierer-in.html>

-> Programmierer Stundenlohn von 24,69 €

Quelle: <https://www.stepstone.de/gehalt/Head-of-IT.html>

-> Head of IT-Stundenlohn von 42,5 €

Quelle: <https://www.test.de/Homeoffice-Pauschale-So-rechnen-Sie-mit-dem-Finanzamt-ab-5788023-0/>

-> Home-office-Pauschale 6 € pro Tag

Abbildung 2: Programmierer Stundenlohn

Abbildung 3: Head of IT-Stundenlohn

Abbildung 4: Home-office-Pauschale

4. Durchführungsphase

4.1. Herstellen der Arbeitsumgebung

1. **Installation von Software und Tools:** Dazu gehören Python, Git, XAMPP (für die lokale Entwicklungsumgebung), Visual Studio Code (oder ein ähnlicher Texteditor) und möglicherweise andere Bibliotheken oder Frameworks, je nach den spezifischen Anforderungen des Projekts.
2. **Einrichten des Projektverzeichnisses:** Es erfolgt die Erstellung eines dedizierten Verzeichnisses für das Projekt auf dem Arbeitscomputer. Die Dateien und Ordner werden in einer logischen Struktur organisiert, um eine einfache Navigation und Auffindbarkeit zu gewährleisten.
3. **Einrichtung der Versionskontrolle:** Um Änderungen am Code zu verfolgen und zu verwalten, wird ein Git-Repository eingerichtet und im Projektverzeichnis initialisiert.
4. **Dokumentation erstellen:** Einrichtung einer Markdown-Datei zum Festhalten von Notizen und Beobachtungen.
5. **Einrichtung von Kommunikationskanälen:** Slack und Asana werden eingerichtet, um die Kommunikation mit anderen Teammitgliedern, dem Kunden und anderen relevanten Parteien zu ermöglichen.

4.2. Eigentlich Umsetzung

4.2.1. Daten von den Webseiten holen

Dieser Prozess gilt für alle 4 Webseiten:

Abbildung 5: Aktivität Diagramm

4.2.1.1 Navigation auf der Webseite mit Python

Ein Python-Skript wird zunächst erstellt, dass die URLs der Haupt- und Unterseiten erfasst und in einem Protokolltextdokument mit dem Namen 'Protokoll.txt' speichert. In dem Protokolltextdokument 'Protokoll.txt' ist auch eine Seitennummerierung enthalten, die mit der erwarteten Anzahl der Seiten verglichen werden kann, die aus der Navigation berechnet werden, kann (**siehe Test 1**).

Abbildung 6: URL-Programmierung

Abbildung 7: Log Output

4.2.1.2 Erstellen von Ordnern für Produkte, PDFs und Bilder

Daraufhin wird eine Ordnerstruktur auf der Festplatte erstellt, wobei für jedes Produkt ein Ordner basierend auf dem Namen der entsprechenden Seite angelegt wird. Etwaige nicht erlaubte Zeichen werden automatisch von Python ersetzt, um die Konsistenz und Integrität der Ordnerstruktur zu gewährleisten. (**siehe Test 2**).

Abbildung 8: Erstellen von Ordnern für Produkte, PDFs und Bilder Programmierung

Abbildung 9: Folders Anzahl

4.2.1.3 Download

Dies gilt für alle Elemente der Seiten (im folgenden Beispiel: h1):

Herunterladen eines Elements von einer Seite

Zunächst wird das Skript nur auf einer Seite getestet, um die URL und den Text des h1-Elements in einer JSON-Datei aufzulisten. Vor dem Loop wird eine Liste mit dem Namen all_values initialisiert. Für jede Seite wird ein Dictionary namens product_info initialisiert, in dem der Wert des h1-Elements unter product_info['h1'] gespeichert wird. Python fügt dieses Dictionary dann der Liste hinzu, und schließlich wird die Liste in eine JSON-Datei geschrieben. Das Protokoll dokument enthält nun die URLs der Seiten sowie das h1-Element der getesteten Seite."

Abbildung 10: Download Programmierung

Abbildung 11: JSON-Output

Abbildung 12: Protokoll

Herunterladen eines Elements von allen Seiten

Wenn das Auflisten des Test-h1-Elements erfolgreich ist, wird das Skript für alle Seiten ausgeführt und im Protokoll dokument vermerkt. Bei auftretenden Problemen wird die URL der Seite im Protokoll dokument angegeben, bei der das Problem aufgetreten ist. Das Skript wird entsprechend angepasst (**siehe Test 3 als Unit Test**). Nach erfolgreicher Ausführung des Skripts auf allen Seiten wird versucht, das nächste Element zu extrahieren.

Nachdem das Skript erfolgreich das h1-Element von allen Seiten extrahiert hat, wird der gleiche Ansatz auf andere Elemente angewendet, einschließlich Beschreibung, Kurzbeschreibung, PDFs, Bilder, Kategorien, Tags usw. Jedes Element wird auf dieselbe Weise im JSON-Format aufgelistet und Bemerkungen im Protokoll dokument vermerkt.

Abbildung 13: JSON

Wenn BeautifulSoup das gesuchte Element nicht finden kann oder falsch extrahiert, wird das Skript auf Selenium umgeschaltet, um eine genauere Suche durchzuführen. Selenium bietet den Vorteil, dass es effektiv ist, wenn Webseiten dynamische Inhalte haben oder JavaScript zur Generierung von Elementen verwendet wird, die von BeautifulSoup möglicherweise nicht erfasst werden. Alle Probleme oder Fehler werden im Protokoll dokument vermerkt, und das Skript wird entsprechend angepasst, um das Problem zu lösen.

Abbildung 14: Protokoll:

Herunterladen aller Produkte (alle Elemente auf allen Seiten)

Sobald das Skript erfolgreich für einzelne Elemente funktioniert, wird es auf alle Seiten angewendet, um die gesamten Elemente zu extrahieren. Dieser Prozess wird systematisch durchgeführt, um sicherzustellen, dass alle relevanten Informationen von jeder Seite erfasst werden. Jegliche auftretenden Probleme werden im Protokoll dokument festgehalten und entsprechende Anpassungen am Skript vorgenommen, um diese zu beheben (**siehe Test 3 als Integration Test**).

Abbildung: 15 JSON-Output

Abbildung 16: Protokoll

Der erste Teil des Projekts ist abgeschlossen. Das Skript, die JSON-Datei und das Protokoll werden im persönlichen Git-Repository gespeichert, das sowohl auf dem Google Drive von Pixelding als auch auf dem Arbeitscomputer von Pixelding und meinem Heimcomputer verfügbar ist. Die PDF- und Bilderordner werden ebenfalls in diesen Repositoryn abgelegt, um eine konsistente und zugängliche Speicherung sicherzustellen.

4.2.2. Daten in lokale WordPress importieren

Dieser Prozess gilt für alle 4 JSON-Dateien:

4.2.2.1 lokale WordPress installieren

Zuerst wird für eine Webseite eine lokale Wordpress-Instanz eingerichtet. Anschließend wird WooCommerce auf dieser lokalen Wordpress-Instanz installiert, gefolgt von der Installation des Plugins WooCommerce Germanized.

4.2.2.2 Daten für WordPress WooCommerce in CSV, und XLSX umbauen

Um die Daten für die Integration in den WordPress WooCommerce-Shop vorzubereiten, wurde ein Python-Skript entwickelt. Dieses Skript liest strukturierte Informationen über Produkte aus einer JSON-Datei und wandelt sie in das erforderliche CSV- und XLSX-Format um. Die JSON-Datei enthält wichtige Details wie Namen, Beschreibungen und andere relevante Informationen zu den Produkten. Während CSV und Excel die gleichen Daten enthalten, bietet Excel aufgrund

seiner tabellarischen Darstellung eine übersichtlichere und benutzerfreundlichere Möglichkeit, die Informationen zu überprüfen und zu bearbeiten.

In dem CSV unter dem Feld 'Typ' bieten sich zwei Auswahlmöglichkeiten an: 'Einfaches Produkt' und 'Variable'. Variable Produkte können entweder den Wert 'Variable' oder 'Variation' haben. Variationen sind, wie der Name schon sagt, Variationen oder unterschiedliche Ausprägungen eines Produkts. Artikelnummern für einfache Produkte und Variation können aus der JSON-Datei eingetragen werden. Für variable Produkte muss eine eindeutige Artikelnummer von Python erstellt werden.

Abbildung 17: Produkt CSV Import Schema

Abbildung 18: Aktivität Diagramm

Skript ausführen für einen Produkt

Das Skript beginnt damit, einen ersten Index der JSON-Daten zu lesen und in die beiden verschiedenen Formate zu konvertieren, um zu testen, ob der Skript funktioniert. (**siehe Test 4 als Unitest**).

Skript ausführen für alle Produkte

Wenn dieser Test erfolgreich ist, wird das Skript auf den gesamten JSON-Datensatz angewendet (**siehe Test 4 als Integration Test**).

Durch die Entwicklung dieses Skripts wurde sichergestellt, dass die Daten für den Online-Shop in den erforderlichen Formaten vorliegen und einfach in das WooCommerce-System integriert werden können. Dies trägt dazu bei, den Prozess der Produktintegration zu rationalisieren und die Effizienz bei der Verwaltung des Online-Shops zu verbessern.

Abbildung 19: CSV Output

4.2.2.3 CSV in WordPress WooCommerce importieren

Dann wird die CSV-Datei mit den Produktdaten über die WordPress WooCommerce-GUI hochgeladen. Im Falle eines Fehlers in der CSV-Datei wird dies hier gemeldet. (**siehe Test 5**) Nach erfolgreicher Integration wird überprüft, ob die Produkte auf der Website ordnungsgemäß dargestellt werden und ob alle Informationen korrekt angezeigt werden. (**siehe Test 6**)

Abbildung 20: WooCommerce Import GUI 1

Abbildung 21: WooCommerce Import GUI 2

Abbildung 22: WooCommerce Import GUI 3

4.2.3. XLSX für Grafiker, SEO, und Kunde anfertigen

Dieser Prozess gilt für alle 4 JSON-Dateien:

4.2.3.1 XLSX-Liste für den Kunden

Der Kunde benötigt eine XLSX-Liste aller Produkte, die entweder als einfaches Produkt oder als Variation definiert sind. Mit dieser Liste kann er Preise, Einkaufspreise und Lagerbestände verwalten. Das gleiche Skript wie oben wird ausgeführt, jedoch mit dem Unterschied, dass Variable-Produkte nicht in die XLSX-Datei geschrieben werden.

4.2.3.2 XLSX-Liste für das SEO-Team

Das SEO-Team benötigt eine XLSX-Liste aller Produkte, um die Beschreibungen und Kurzbeschreibungen anzupassen. Dabei können sie auch die Parameter von Variable-Produkten anpassen, falls erforderlich. Diese XLSX-Datei wurde zusammen mit der CSV-Generierung automatisch erstellt.

4.2.3.3 XLSX-Liste für die Grafiker

Das Grafiker-Team benötigt ebenfalls eine XLSX-Liste aller Produkte, um die Bilder zu verwalten. Sie können die Bildpfade hier in Bezug auf die Ordnerstruktur ändern. Die Grafiker müssen die neuen Bilder manuell in die entsprechenden Ordner hochladen. Diese XLSX-Datei wurde zusammen mit der CSV-Generierung automatisch erstellt.

4.2.4. Daten in lokale WordPress updaten

4.2.4.1 Erstellen einer neuen lokale WordPress-WooCommerce-Instanz

Für die Aktualisierung der Daten in WordPress WooCommerce wird zunächst eine neue lokale WordPress-WooCommerce-Instanz erstellt. Die 4 ursprünglichen CSV-Dateien werden erneut geladen. Die vorherigen WordPress-WooCommerce-Instanzen dienen als Backup und können zum Vergleich herangezogen werden.

4.2.4.2 Konvertierung der XLSX-files zu CSV-files mit Python

Wenn SEO, Grafiker, und Kunden ihre XLSX-Files geupdatet haben, geht es weiter. Die vier XLSX-Dateien des Kunden werden in vier CSV-Dateien mit Python umgewandelt. Dasselbe wird für die vier XLSX-Dateien des SEO-Teams und die vier XLSX-Dateien des Grafikteams durchgeführt. Mithilfe von Python werden nur die erforderlichen Spalten exportiert. Die Artikelnummern werden als Zeichenkette exportiert, wobei führende Nullen beibehalten werden. Die Dezimalzahlen für Verkaufs- und Einkaufspreise werden im Export durch einen Punkt anstelle eines Kommas dargestellt. Die deutschen Buchstaben werden bei der Kodierung mit 'utf-8' behandelt (**siehe Test 7**).

Abbildung 23: XLSX zu CSV

4.2.4.3 Hochladen der CSV in den neuen Lokalen WordPress

Die zwölf CSV-Dateien werden dann über die WooCommerce-GUI importiert, wobei die Option "Aktualisieren" ausgewählt wird (**siehe test5**). Nach erfolgreicher Prüfung kann die lokale WordPress-WooCommerce-Instanz auf den Online-Shop des Kunden übertragen werden (**siehe Test 6**).

4.2.5. Daten in Onlineshop hochladen

4.2.5.1 Exportieren der CSV-File aus dem Lokal Shop

Um die Daten in den Online-Shop zu integrieren, werden zunächst die Produktinformationen aus dem lokalen Shop über die WooCommerce GUI als CSV exportiert. Dabei werden die URLs für Bilder und PDFs noch lokal referenziert.

4.2.5.2 Anpassen der Bilder und PDF URL in der CSV-File

Anschließend werden diese CSV-Dateien in Visual Studio Code geöffnet, um die lokalen URLs automatisch durch die Online-URLs des Online-Shops zu ersetzen.

4.2.5.3 Onlinelagerung der Bilder und PDF

Die Bilder und PDFs werden dann mit FileZilla (FTP) auf den Server des Kunden hochgeladen, wo WooCommerce sie später finden kann.

4.2.5.4 Import der CSV-File in den Onlineshop

Nachdem der Online-Shop in den Bearbeitungsmodus versetzt wurde, erfolgt der Import der CSV-Dateien in WooCommerce über die GUI. WooCommerce lädt automatisch alle Produkte hoch. Die Bilder und PDFs werden gefunden und an den richtigen Orten referenziert. (**siehe Test 5 als Abnahmetest**). Abschließend wird überprüft, ob alle Daten korrekt hochgeladen und im Online-Shop angezeigt werden. Nach Abschluss der Überprüfung wird der Bearbeitungsmodus des Online-Shops deaktiviert. (**siehe Test 6 als Abnahmetest**).

Speichern in GitHub, Google und auf den Computer immer wieder!

4.3. Testen

Tests spielen eine entscheidende Rolle in der Softwareentwicklung und bieten einen Mechanismus zur Bewertung der Funktionalität und Zuverlässigkeit von Code. Obwohl Tests keine fehlerfreie Leistung in allen Situationen garantieren können, bleiben sie das effektivste Mittel, um potenzielle Probleme vor der Bereitstellung zu identifizieren und zu beheben. Im Folgenden werden die während des oben beschriebenen Programmierprozesses durchgeführten Tests aufgelistet.

Test 1: Es wird überprüft, ob alle Seiten erfolgreich gelesen werden können, um sicherzustellen, dass alle relevanten Informationen erfasst werden. Dabei wird auch die Gesamtanzahl der Seiten ermittelt, indem die Anzahl der Seiten in der Navigation gezählt und mit der Anzahl der gesuchten und nummerierten Seiten im Protokoll verglichen wird.

Test 2: Es wird überprüft, ob die Anzahl der erstellten Ordner der Anzahl der erfassten Seiten entspricht. Dabei wird sicherzustellen versucht, dass jeder Produktordner basierend auf dem Namen der entsprechenden Seite korrekt erstellt wurde. Des Weiteren wird überprüft, ob die Namen der Ordner gültig sind und nur erlaubte Zeichen enthalten.

Test 3: Im Protokolldokument wird unter der Seitennummer und der URL eine neue Zeile mit dem gesuchten Element hinzugefügt. Es wird vermerkt, ob das Element gefunden wurde oder nicht. Das Python-Skript wird in einer Schleife ausgeführt. Innerhalb dieser Schleife wird ein if-else- oder Try-Except-Block eingerichtet, um sicherzustellen, dass das Skript auch dann ausgeführt wird, wenn ein Element nicht gefunden wird. Die Struktur des Protokolls zeigt gefundene Elemente nach einer Folge Minuszeichen (-) und nicht gefundene Elemente nach Pluszeichen (+).

Abbildung 16: Protokoll

Test 4: Dieses Testverfahren besteht darin, manuell visuell zu überprüfen, ob die Daten im Excel-Dokument in die richtigen Spalten übertragen wurden und ob die Variationen und Einheiten korrekt übernommen wurden.

Test 5: Dieses Testverfahren erfolgt während des Hochladens der CSV-Datei über das Word-Press WooCommerce-GUI-Upload-Formular. Dabei werden Fehler in der CSV-Datei gemeldet, entweder während des Hochladens oder direkt nach Abschluss des Vorgangs. Es wird auch die Anzahl der importierten Produkte angezeigt, sodass sie mit der Anzahl der Zeilen aus der Excel-Datei verglichen werden kann.

Abbildung 24: WooCommerce Fehler Meldung

Test 6: Nach der erfolgreichen Integration wird jede Seite überprüft, um sicherzustellen, dass die Produkte ordnungsgemäß dargestellt werden und dass alle Informationen korrekt angezeigt werden. Dazu wird jede Seite einzeln aufgerufen und mit ihrer ursprünglichen URL verglichen. Beim Lokal Test wird mit den Seiten des Herstellers verglichen, beim Abnahmetest wird mit den Seiten des Lokalshops verglichen. Außerdem werden alle Varianten-Dropdown-Menüs angeklickt, um sicherzustellen, dass sie ordnungsgemäß funktionieren.

Test 7: Hierbei geht es darum, zu überprüfen, ob die deutschen Buchstaben ordnungsgemäß behandelt werden und ob die Dezimalzahlen für Verkaufs- und Einkaufspreise im Export durch einen Punkt anstelle eines Kommas dargestellt werden.

5. Übergabe und Auswertung

5.1. Projektübergabe

Die neuen Produkte wurden erfolgreich in den Online-Shop integriert. Die Dokumentation und Kopien des Codes, der Bilder und der PDF befinden sich auf den Computern und Google Drive von Pixelding sowie auf GitHub.

Die Excel-Dateien sind ebenfalls auf Google Drive gespeichert, und der Kunde kann bei Bedarf seine Preise ändern. Die SEO-Abteilung kann die Beschreibung und Kurzbeschreibung monatlich für SEO-Zwecke aktualisiert werden.

Zu diesem Zweck kann der Python-Code zum Konvertieren von Excel- in CSV-Dateien verwendet werden.

5.2. Mitarbeiterschulung, Bedienungsanleitung

Es wurde eine Schulung durchgeführt in der Funktionsweise des Project vorgestellt wurde.

5.3. Soll-Ist-Vergleich (inhaltlich, Zeitlich, kostentechnisch)

Inhaltlich:

Im Hinblick auf die inhaltliche Umsetzung des Projekts können wir feststellen, dass die ursprünglichen Ziele erfolgreich erreicht wurden. Die Python-Skripte wurden entwickelt, um Daten von den

Webseiten der Hersteller herunterzuladen, ohne eine definierte Kommunikation über beispielsweise Schnittstellen. Diese Daten wurden anschließend in JSON-Dateien referenziert und organisiert, die als Grundlage zur Herstellung der CSV- und XLSX-Dokumente für Kunden, Grafiker und SEO dienten. Die CSV-Dateien konnten erfolgreich in den WooCommerce Online-Shop importiert werden. Darüber hinaus wurden Tests durchgeführt, um sicherzustellen, dass die Skripte ordnungsgemäß funktionieren und die Daten korrekt verarbeitet werden. Die Überprüfung der Webseite nach der Integration hat gezeigt, dass die Produkte ordnungsgemäß dargestellt werden und alle Informationen korrekt angezeigt werden. Insgesamt ist das Projekt inhaltlich erfolgreich umgesetzt worden.

Zeitlich:

Alle Punkte konnten pünktlich abgearbeitet werden außer dem Download-Prozess. Der beanspruchte mehr Zeit als erwartet, bedingt durch die Dynamik der generierten Inhalte, die nicht direkt von BeautifulSoup erfasst werden konnten. Der Übergang zu Selenium stellte anfangs eine kleine Herausforderung dar, erwies sich jedoch als äußerst lohnend. Trotz des anfänglichen Zeitverlustes konnte dieser später durch eine effiziente Vorgehensweise und zügige Fortschritte in den nachfolgenden Schritten wieder aufgeholt werden.

Kostentechnisch:

Die Kosten für das Projekt blieben im Rahmen des Budgets. Es waren keine zusätzlichen Ressourcen oder Ausgaben erforderlich, um das Projekt erfolgreich abzuschließen. Die Kosten wurden effizient verwaltet und alle Ressourcen wurden sorgfältig genutzt, um die Ziele des Projekts zu erreichen. Insgesamt wurden die Kosten für das Projekt gut kontrolliert und blieben im geplanten Rahmen.

5.4. Technische Auswertung und Ausblick

Die Arbeit an diesem Projekt war äußerst lohnend. Die Implementierung des Python-Skripts zur Datenverarbeitung hat gezeigt, wie effizient und zeitsparend Automatisierung sein kann. Der Ausblick für das Programm ist vielversprechend. In Zukunft könnte es weiterentwickelt werden, um zusätzliche Funktionen und Verbesserungen zu integrieren, wie beispielsweise eine benutzerfreundliche Oberfläche für den CSV Excel Konvertierung.

5.5. Persönliche Auswertung

Die Zeit verging wie im Flug, und ich habe es wirklich genossen, an diesem Projekt zu arbeiten. Es war eine großartige Gelegenheit, meine Fähigkeiten in Python-Programmierung und Datenverarbeitung zu verbessern und gleichzeitig einen konkreten Beitrag zum Unternehmen zu leisten.

Abbildung 1: Aktivität Diagramm

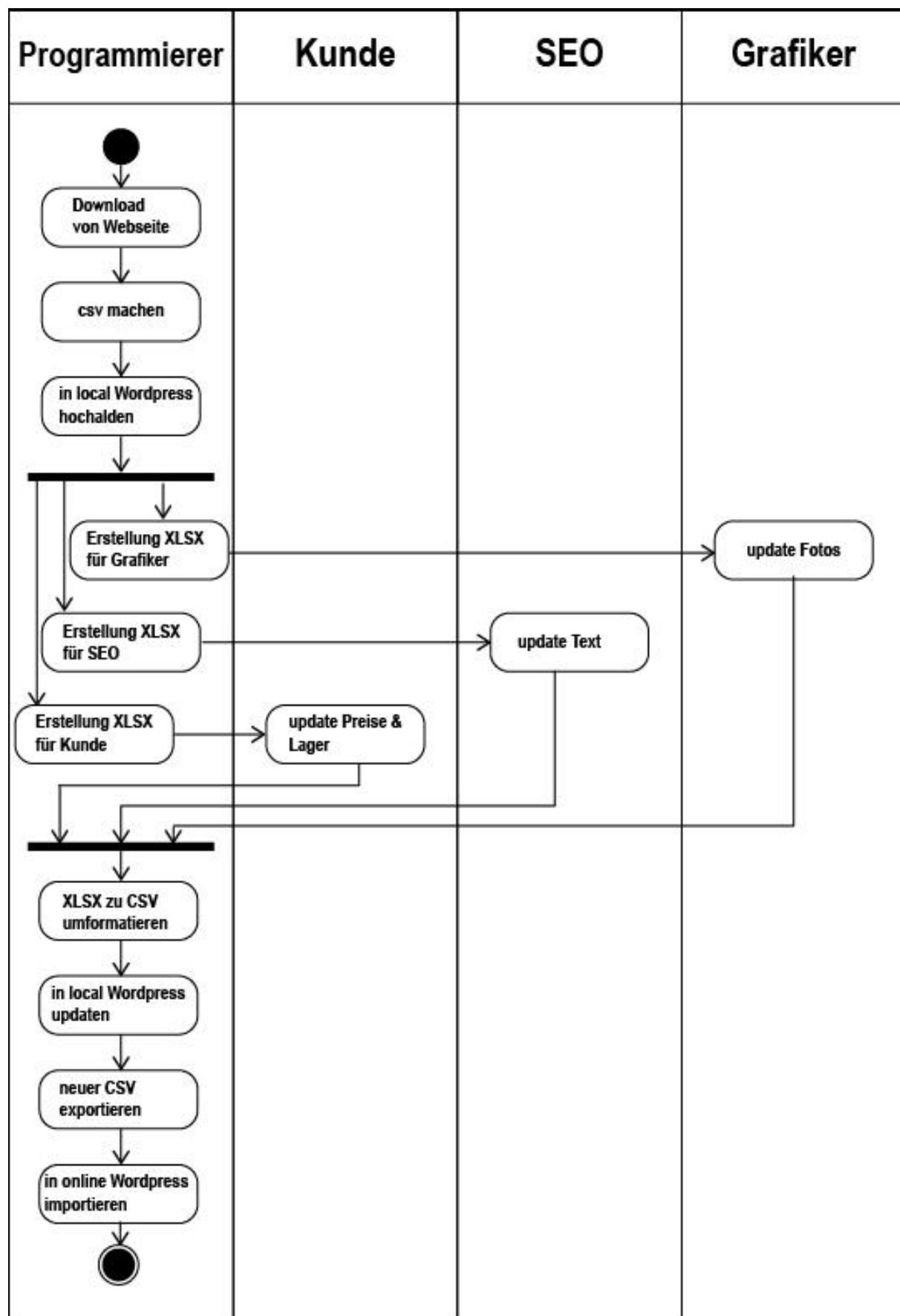


Abbildung 2: Programmierer Stundenlohn



Abbildung 3: Head of IT-Stundenlohn

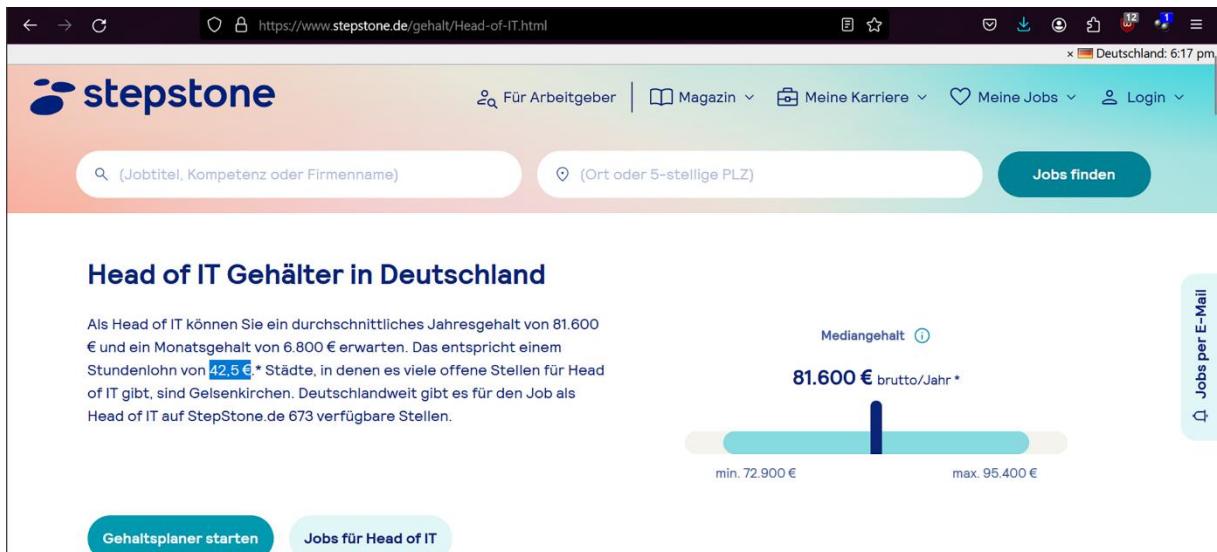


Abbildung 4: Home-office-Pauschale



Das sollten Sie wissen

- Die 2020 eingeführte Homeoffice-Pauschale galt zunächst nur befristet, doch mittlerweile gibt es sie **dauerhaft**.
- Die Pauschale gilt für **alle Berufstätigen** – egal, ob angestellt oder selbstständig. Auch für jeden Ehepartner einzeln.
- Ihre Angaben machen Sie in der **Anlage N** Ihrer Steuererklärung.
- Die Pauschale zählt zu den [Werbungskosten](#). Sie sparen erst Steuern, wenn Sie im Jahr mit all Ihren beruflichen Ausgaben den Werbungskosten-Pauschbetrag überspringen. Dieser beträgt aktuell **1 230 Euro**.
- **Seit Anfang 2023 rechnen Sie für bis zu 210 Homeoffice-Tage jeweils 6 Euro ab.** Sitzen Sie noch an der Steuererklärung für **2022**, machen Sie jeweils **5 Euro** für bis zu **120 Tage** geltend.

Abbildung 5: Aktivität Diagramm

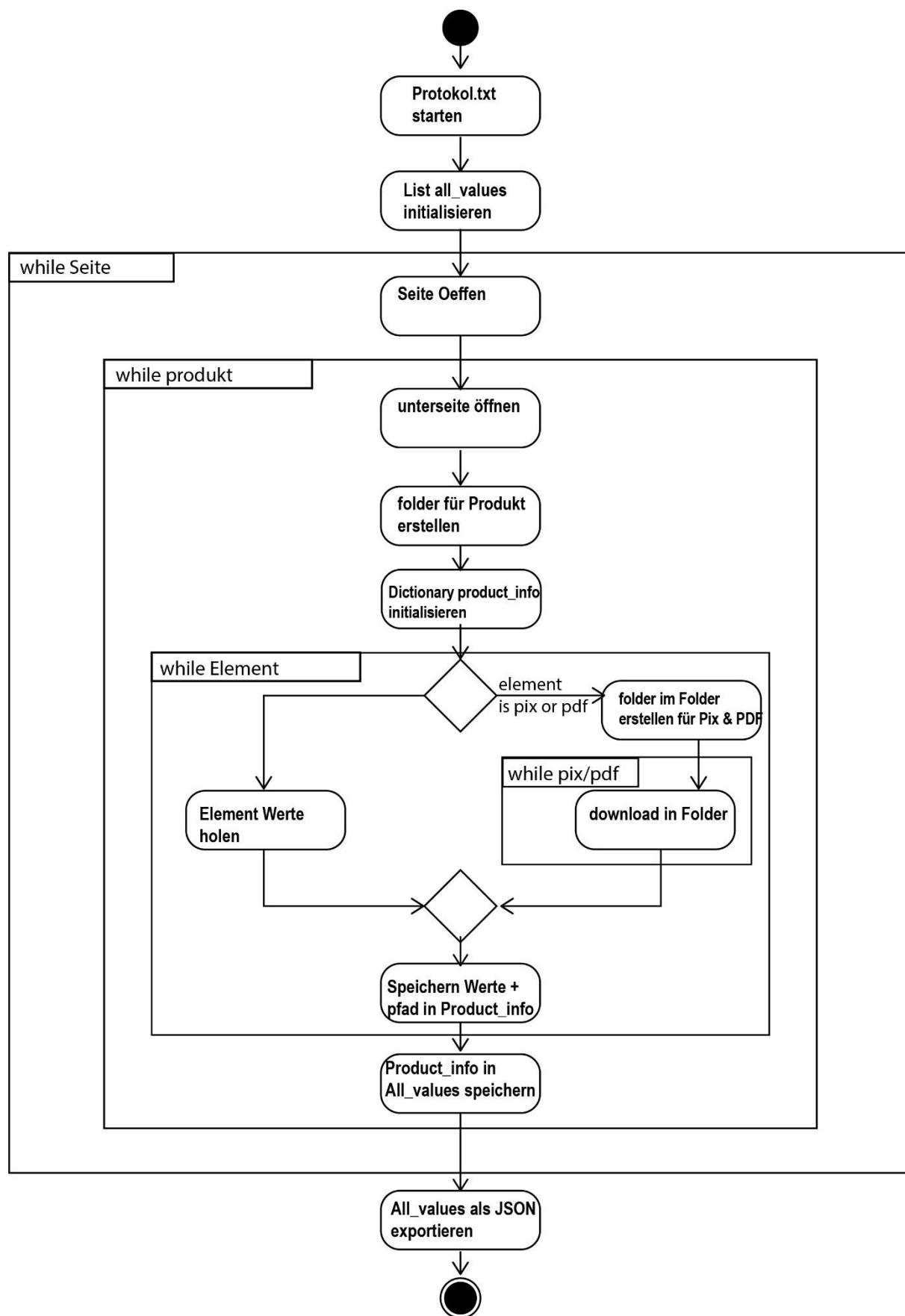


Abbildung 6: URL-Programmierung

Abbildung 7: Log Output

```
main page 1 from 13
-----
url: https://shop.buzil.de/anwendungsbereich/?p=1&f=

page 1 from 145
- found https://shop.buzil.de/spezial/air-provence-g-565

page 2 from 145
- found https://shop.buzil.de/kuechen-/lebensmittelhygiene/bistro-g-435

page 3 from 145
- found https://shop.buzil.de/sanitaer/bucasan-clear-g-463

page 4 from 145
- found https://shop.buzil.de/sanitaer/bucasan-fresh-p-920

page 5 from 145
- found https://shop.buzil.de/sanitaer/bucasan-hit-p-921

page 6 from 145
- found https://shop.buzil.de/sanitaer/bucasan-sanibond-g-457

page 7 from 145
- found https://shop.buzil.de/sanitaer/bucasan+-intense-p-922

page 8 from 145
- found https://shop.buzil.de/spezial/buz-air-ambra-g-564

page 9 from 145
- found https://shop.buzil.de/kuechen-/lebensmittelhygiene/buz-calc-ex-g-437

page 10 from 145
- found https://shop.buzil.de/spezial/buz-fresh-magic-g-567

page 11 from 145
- found https://shop.buzil.de/oberflaeche/buz-glass-p-912

...
```

Abbildung 8: Erstellen von Ordnern für Produkte, PDFs und Bilder Programmierung

```
# create folder with that name
invalid_chars = ['<', '>', ':', '"', '/', '\\', '|', '?', '*']

# Extract the name from the href
folder_name = os.path.basename(product_info['a'])

# Ersetze ungültige Zeichen durch Unterstriche
for char in invalid_chars:
    folder_name = folder_name.replace(char, '_')

# Create a folder with the name from the href
folder_path = os.path.join(os.getcwd(), folder_name)
os.makedirs(folder_path, exist_ok=True)
custom_print(f'-- making folder {folder_path}'')
```

Abbildung 9: Folders Anzahl

12kg-h-645	bucasan-pine-g-453	buz-dish-fix-g-530
air-provence-g-565t	bucasan-sanibond-g-457	buz-finesse-g-542
aktiv-g-433	bucasan-sanibreeze-g-454	buz-flow-g-577
auslaufhahn-10-l-laundry-lh-630	bucasan-trendy-t-464	buz-fresh-magic-g-567
auslaufhahn-20-l-laundry-lh-620	bucazid-s-g-467	buz-glass-p-912
auslaufhahn-h-620	budenat-alkasept-d-445	buz-grillmaster-g-575
bistro-g-435	budenat-intense-d-443	buz-grillmaster-ready-to-use-g-576
blitz-citro-g-481	budenat-lm-d-447	buz-laundry-bright-l-832
blitz-orange-g-482	budenat-protect-hand-d-807	buz-laundry-chlorin-l-833
blitz-tropic-g-483	budenat-rapid-d-444	buz-laundry-color+-l-821
bucalex-g-460	budenat-rapid-effect-d-446	buz-laundry-compact-l-810
bucal-g-468	buz+-intense-p-911	buz-laundry-enz-3-l-820
bucasan+-intense-p-922	buz-air-ambra-g-564	buz-laundry-soft-l-830
bucasan-clear-g-463	buz-calc-ex-g-437	buz-laundry-strong-l-831
bucasan-fresh-p-920	buz-contracalc-g-461	buz-leather-c-580
bucasan-hit-p-921	buz-defoam-g-478	buz-mark-ex-g-559

Abbildung 10: Download Programmierung

```
#... rest of code
# todo linked page
# Make an HTTP GET request to the linked page
response2 = requests.get(a_tag['href'])
# Check if the request was successful
if response2.status_code == 200:
# Parse the HTML content
    soup2 = BeautifulSoup(response2.text, 'html.parser')
    # Extract all <div> elements with the specified ID
    section = soup2.find('div', id='contentwrap')
    #
    # | | / |
    # | |_`| |
    # | '_\| |
    # | ||||| |
    # |_||_|_\|/
    # Extract the h1 element within the section
    h1_element = section.find('h1')
    if h1_element:
        h1_text = h1_element.get_text().strip()
        # strip is trimming empty space
        product_info['h1'] = h1_text
        custom_print(f'-- FOUND H1')
    else:
        product_info['h1'] = ''
        custom_print(f'+ + h1 is missing ?')
    # -----
    # write to all values
    all_values.append(product_info)
#
# ( )
#
# | | / | / | \ | ' | \
# | \_ \ ( ) | | | |
# | | / \_ / | | | |
# | / |
# | | /
# Specify the directory path where the JSON file is located
directory_path = ''
# Specify the file names
json_file_name = f'buzil_{current_datetime}.json'
# Save all_values as a JSON file for each URL
with open(json_file_name, 'w', encoding='utf-8') as json_file:
    json.dump(all_values, json_file, ensure_ascii=False, indent=4)
print("JSON file page created.")
json_file_path = os.path.join(directory_path, json_file_name)
```

Abbildung 11: JSON-Output

```
[  
  {  
    "a": "https://shop.buzil.de/spezial/air-provence-g-565t",  
    "h1": "Air Provence"  
  }  
]
```

Abbildung 12: Protokoll

```
main page 1 from 13  
-----  
url: https://shop.buzil.de/anwendungsbereich/?p=1&f=  
  
page 1 from 145  
- found https://shop.buzil.de/spezial/air-provence-g-565t  
- - making folder C:\Users\greg\Documents\sites\project\buzil\air-provence-g-565t  
- - - FOUND H1
```

Abbildung 13: JSON

```
[  
  {  
    "a": "https://shop.buzil.de/spezial/air-provence-g-565t",  
    "h1": "Air Provence"  
  },  
  {  
    "a": "https://shop.buzil.de/kuechen-/lebensmittelhygiene/bistro-g-  
435",  
    "h1": "Bistro"  
  },  
  {  
    "a": "https://shop.buzil.de/sanitaer/bucasan-clear-g-463",  
    "h1": "Bucasan® Clear"  
  },  
  {  
    "a": "https://shop.buzil.de/sanitaer/bucasan-fresh-p-920",  
    "h1": "Bucasan® Fresh"  
  },  
  ...
```

Abbildung 14: Protokoll

main page 1 from 13

url: <https://shop.buzil.de/anwendungsbereich/?p=1&f=>

page 1 from 145

- found <https://shop.buzil.de/spezial/air-provence-g-565>
- making folder C:\Users\greg\Documents\sites\project\buzil\air-provence-g-565t
- FOUND H1

page 2 from 145

- found <https://shop.buzil.de/kuechen-/lebensmittelhygiene/bistro-g-435>
- making folder C:\Users\greg\Documents\sites\project\buzil\bistro-g-435
- FOUND H1

page 3 from 145

- found <https://shop.buzil.de/sanitaer/bucasan-clear-g-463>
- making folder C:\Users\greg\Documents\sites\project\buzil\bucasan-clear-g-463
- FOUND H1

page 4 from 145

- found <https://shop.buzil.de/sanitaer/bucasan-fresh-p-920>
- making folder C:\Users\greg\Documents\sites\project\buzil\bucasan-fresh-p-920
- FOUND H1

page 5 from 145

- found <https://shop.buzil.de/sanitaer/bucasan-hit-p-921>
- making folder C:\Users\greg\Documents\sites\project\buzil\bucasan-hit-p-921
- FOUND H1

...

Abbildung 15: JSON-Output

```
[  
  {  
    "a": "https://shop.buzil.de/spezial/air-provence-g-565t",  
    "h1": "Air Provence G 565T",  
    "image_urls": [  
      "https://shop.buzil.de/media/image/0a/1a/ca/2Q0Ln7huZ4R9d01_1280x1280.jpg",  
      "https://shop.buzil.de/media/image/c8/8b/07/2Q0Ln7huZ4R9d01_200x200.jpg",  
      "https://shop.buzil.de/media/image/7e/d6/56/2Q0Ln7huZ4R9d01.jpg",  
      "https://shop.buzil.de/media/image/5d/0e/77/2Q0Ln7huZ4R9d01_600x600.jpg",  
      "https://shop.buzil.de/media/image/3b/f1/21/2Q0Ln7huZ4R9d01_600x600@2x.jpg"  
    ],  
    "image_local": [  
      "air-provence-g-565t/img/2Q0Ln7huZ4R9d01_1280x1280.jpg",  
      "air-provence-g-565t/img/2Q0Ln7huZ4R9d01_200x200.jpg",  
      "air-provence-g-565t/img/2Q0Ln7huZ4R9d01.jpg",  
      "air-provence-g-565t/img/2Q0Ln7huZ4R9d01_600x600.jpg",  
      "air-provence-g-565t/img/2Q0Ln7huZ4R9d01_600x600@2x.jpg"  
    ],  
    "product_detail_description_section": [  
      {  
        "header": "Gebrauchsfertiger Raumspray mit Aktiv-Geruchsblocker",  
        "content": "In Wohn- und Geschäftsräumen, öffentlichen Einrichtungen wie Krankenhäusern, Altenheimen sowie in der Gastronomie, Sanitärs- und WC-Anlagen, Aufenthalts- und Umkleideräumen, Sport- und Wellnessbereichen etc.; für die schnelle und langanhaltende Geruchsvernichtung und Luftverbesserung"  
      },  
      {  
        "header": "Produktvorteile",  
        "content": [  
          "Aromatischer Duft, der an frisch geernteten Lavendel erinnert sorgt für eine angenehme Raumatmosphäre",  
          "Neutralisiert schnell unangenehme Gerüche mit Aktiv-Geruchsblocker",  
          "Hinterlässt keine Flecken",  
          "Umweltfreundlicher Zerstäuber ohne Treibgas"  
        ]  
      }  
    ],  
    ...  
  }]
```

Abbildung 16: Protokoll

```
main page 1 from 13
```

```
-----  
url: https://shop.buzil.de/anwendungsbereich/?p=1&f=
```

```
page 1 from 145
```

```
- found https://shop.buzil.de/spezial/air-provence-g-565t
-- making folder C:\Users\greg\Documents\sites\project\buzil\air-provence-g-565t
--- FOUND H1
--- DESCRIPTION-DATA
---- downloading description_data: 1
---- downloading description_data: 2
--- ADDITIONAL DATA
---- downloading additionail detail: 1
---- downloading additionail detail
---- downloading additionail detail: 2
---- downloading additionail detail
--- HINT
--- DOWNLOADING PRODUKKT DOWNLOAD TITLE 1
--- PRODUKT DOWNLOAD CONTENT
---- product_download_content found
---- product_download_content li found: 1
---- product_download li
----- downloading text and pdf for Betriebsanweisungen
----- downloading PDF for Betriebsanweisungen
---- product_download_content li found: 2
...
```

Abbildung 17: Produkt CSV Import Schema

Table of Contents

- [General CSV guidelines](#)
- [CSV columns and formatting](#)

This is the schema used for the built in [CSV importer and exporter](#) in WooCommerce 3.1+.

Visit the [Product CSV Importer and Exporter](#) guide for more information on how to use it.

General CSV guidelines

- CSVs should be in UTF-8 format.
- Any dates should be defined for the store's local timezone.
- If importing a boolean value (true or false) use 1 or 0 in your CSV.
- Multiple values in a field get separated with commas.
- If referencing an existing product ID, prefix the id with `id:`. If referencing a SKU, no prefix is needed. e.g. `id:100, SKU101`
- Custom meta is supported/imported but only plain text - no support for JSON or Serialised data.
- Taxonomy term hierarchy is denoted with `>`, terms are separated with commas.
- Serialised data is not supported.
- Images need to be pre-uploaded or available online in order to import to your store.
- External URLs are supported and imported into the media library if used.
- You can define the filename if the image already exists in the media library.

CSV columns and formatting

CSV Column Name	Maps to product property	Example	Notes
ID	id	100	Defining this will overwrite data for that ID on import.
Type	type	simple, variation, virtual	Product Type. Valid values: simple, variable, grouped, external, variation, virtual, downloadable.

			Multiple types can be used CSV separated.
SKU	sku	my-sku	Required. Auto-generated if missing.
Name	name	My Product Name	Required.
Published	status	1	1 for published, 0 for private, -1 for draft.
Is featured?	featured	1	1 or 0
Visibility in catalog	catalog_visibility	visible	Supported values: <code>visible</code> , <code>catalog</code> , <code>search</code> , <code>hidden</code>
Short description	short_description	This is a product.	
Description	description	This is more information about a product.	
Date sale price starts	date_on_sale_from	2013-06-07	Date (start of day) or leave blank.
Date sale price ends	date_on_sale_to	2013-06-07	Date (end of day) or leave blank.
Tax status	tax_status	taxable	Supported values: <code>taxable</code> , <code>shipping</code> , <code>none</code>
Tax class	tax_class	standard	Can use any existing tax class.
In stock?	stock_status	1	1 or 0
Stock	manage_stock / <small>parent</small>	20	Numeric stock level enables stock management. <code>parent</code> can be

	stock_quantity		used for variations. Blank = no stock management.
Low stock amount	low_stock_amount	3	Empty or a number
Backorders allowed?	backorders	1	1, 0, or notify
Sold individually?	sold_individually	1	1 or 0
Weight (unit)	weight	100	Parse only numbers.
Length (unit)	length	20	Parse only numbers.
Width (unit)	width	20	Parse only numbers.
Height (unit)	height	20	Parse only numbers.
Allow customer reviews?	reviews_allowed	1	1 or 0
Purchase Note	purchase_note	Thanks for buying it buddy.	
Sale price	sale_price	20.99	Sale price.
Regular price	regular_price	24.99	Regular price.
Categories	category_ids	Category 1, Category 1 > Category 2	CSV list of categories. > used for hierarchy.
Tags	tag_ids	Tag 1, Tag 2	CSV list of tags.
Shipping class	shipping_class_id	Name	Name of shipping class
Images	image_id /	http://somewhere.com/image.jpg http://	First is featured

gallery_image_ids	image
Download limit	n/a or a limit.
Download expiry days	n/a or a day limit.
Parent	Set parent ID. Used for variations. Can be just a numeric ID e.g. id:100 or a SKU. Export will use SKU when possible.
Grouped products	List of IDs. Can be just a numeric ID e.g. id:100 or a SKU. Export will use SKU when possible.
Upsells	List of IDs. Can be just a numeric ID e.g. id:100 or a SKU. Export will use SKU when possible.
Cross-sells	List of IDs. Can be just a numeric ID e.g. id:100 or a SKU. Export will use SKU when possible.
External URL	Product external URL.
Button text	Custom product "buy" button.

Many orders used

Position	menu_order	1	menu_order, used for sorting.
Attribute 1 name	attributes	Color	Looks for global attribute or uses text if not found. Include as many as needed. "Used for variations" is set automatically.
Attribute 1 value(s)	attributes	Blue, Red, Green	List of values. Variations only need 1 value. First is used if multiple get provided.
Attribute 1 default	default_attributes	Blue	Default value for variable products.
Attribute 1 visible	attributes	1	1 or 0. Mapping screen labels this as "Attribute Visibility"
Attribute 1 global	attributes	1	1 or 0. Mapping screen labels this as "Is a global attribute?"
Download 1 name	downloads	Download 1	
Download 1 URL	downloads	url.zip	

Abbildung 18: Aktivität Diagramm

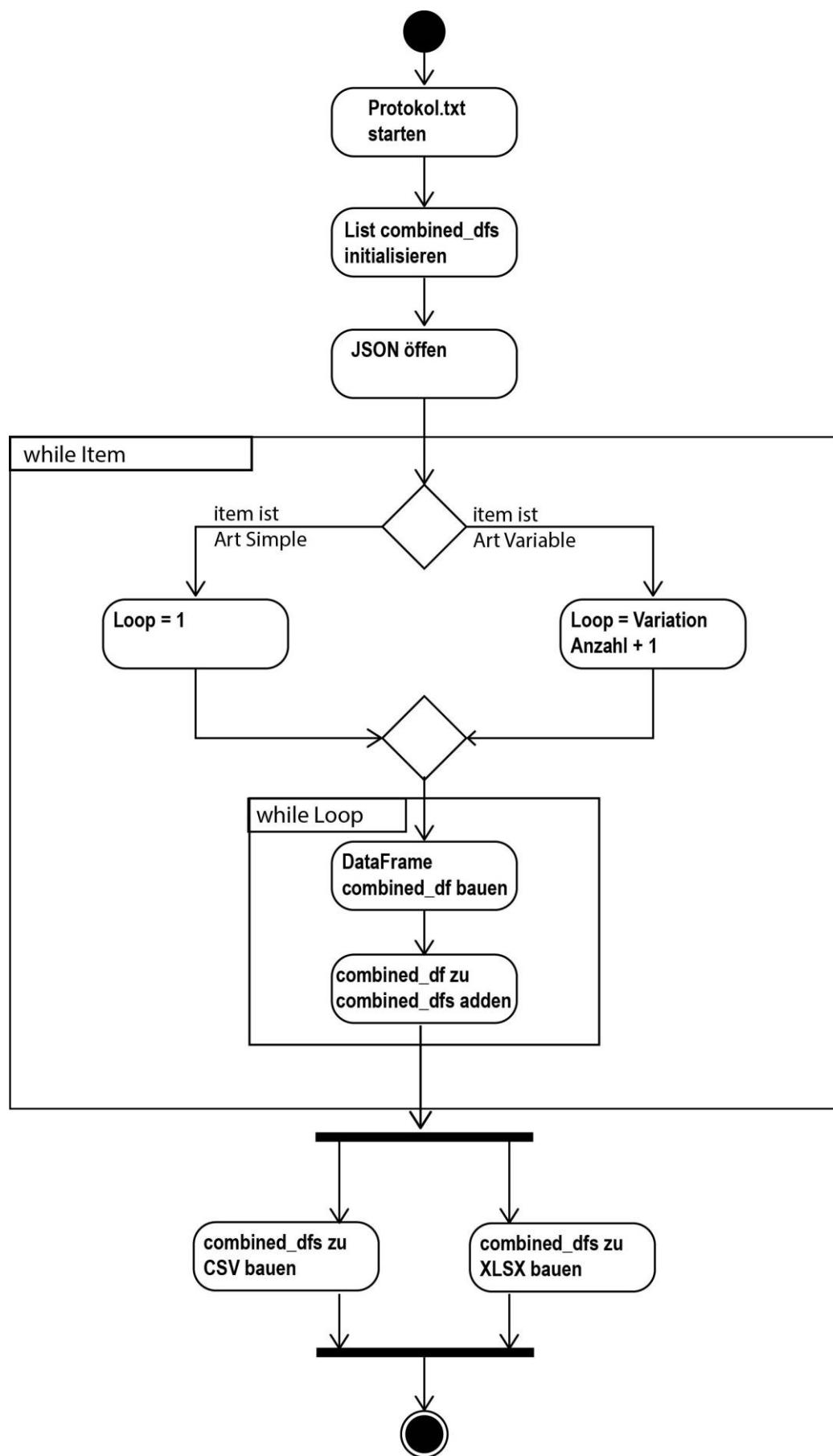


Abbildung 19: CSV Output

```
Typ,Artikelnummer,Name,Veröffentlicht,"Ist hervorgehoben?", "Sichtbarkeit im Katalog",Kurzbeschreibung,Bes
variable,sku_blau-022,"Stretch Light",1,0,visible,"<h3>Mikrofaser Stretch Light Tuch</h3>\n<p>Leistungsst
variable,sku_blau-003,"Hochleistungstuch Platin",1,0,visible,"<h3>Hochwertiges und volumiges Mikrofaseru
variable,sku_blau-093,Schwammtuch,1,0,visible,"<h3>Mikrofaser Schwammtuch</h3>\n<p>Leistungsstarkes Reini
variable,sku_blau-076,Softtuch,1,0,visible,"<h3>Mikrofaser Softtuch</h3><p> mit feinen Fasern für empfindlic
variable,sku_blau-085,Borstentuch,1,0,visible,"<h3>Mikrofaser Softtuch</h3>\n<p>Feinste Fasern für empfindliche
variable,sku_blau-015,"Polier- & Geschirrtuch",1,0,visible,"<h3>Mikrofaser Gläsertuch mit filigraner Struktur
variation,blau-022,"Stretch Light - blau-022",1,0,visible,,,,taxable,parent,1,800,10,0,0,0,40,40,0,0,,,"1
variation,rosa-021,"Stretch Light - rosa-021",1,0,visible,,,,taxable,parent,1,700,10,0,0,0,40,40,0,0,,,"1
variation,gelb-024,"Stretch Light - gelb-024",1,0,visible,,,,taxable,parent,1,500,10,0,0,0,40,40,0,0,,,"1
variation,grün-023,"Stretch Light - grün-023",1,0,visible,,,,taxable,parent,1,100,10,0,0,0,40,40,0,0,,,"1
variation,weiß-075,"Stretch Light - weiß-075",1,0,visible,,,,taxable,parent,0,0,10,0,0,0,40,40,0,0,,,"1
variation,schwarz-106,"Stretch Light - schwarz-106",1,0,visible,,,,taxable,parent,1,30,10,0,0,0,40,40,0,0,,,"1
variation,blau-003,"Hochleistungstuch Platin - blau-003",1,0,visible,,,,taxable,parent,1,40,10,0,0,0,40,40,0,0,,,"1
variation,rosa-072,"Hochleistungstuch Platin - rosa-072",1,0,visible,,,,taxable,parent,1,35,10,0,0,0,40,40,0,0,,,"1
variation,gelb-001,"Hochleistungstuch Platin - gelb-001",1,0,visible,,,,taxable,parent,1,30,10,0,0,0,40,40,0,0,,,"1
variation,grün-002,"Hochleistungstuch Platin - grün-002",1,0,visible,,,,taxable,parent,1,40,10,0,0,0,40,40,0,0,,,"1
variation,blau-093,"Schwammtuch - blau-093",1,0,visible,,,,taxable,parent,0,0,10,0,0,0,17,23,0,0,,,"1,4
variation,rosa-094,"Schwammtuch - rosa-094",1,0,visible,,,,taxable,parent,0,0,10,0,0,0,17,23,0,0,,,"1,4
Variationen: 2016 von 2016 "Schwammtuch - blau-093" 1,0,visible taxable parent 0,0,10,0,0,0,17,23,0,0,,,"1,4"
```

Abbildung 20: WooCommerce Import GUI 1

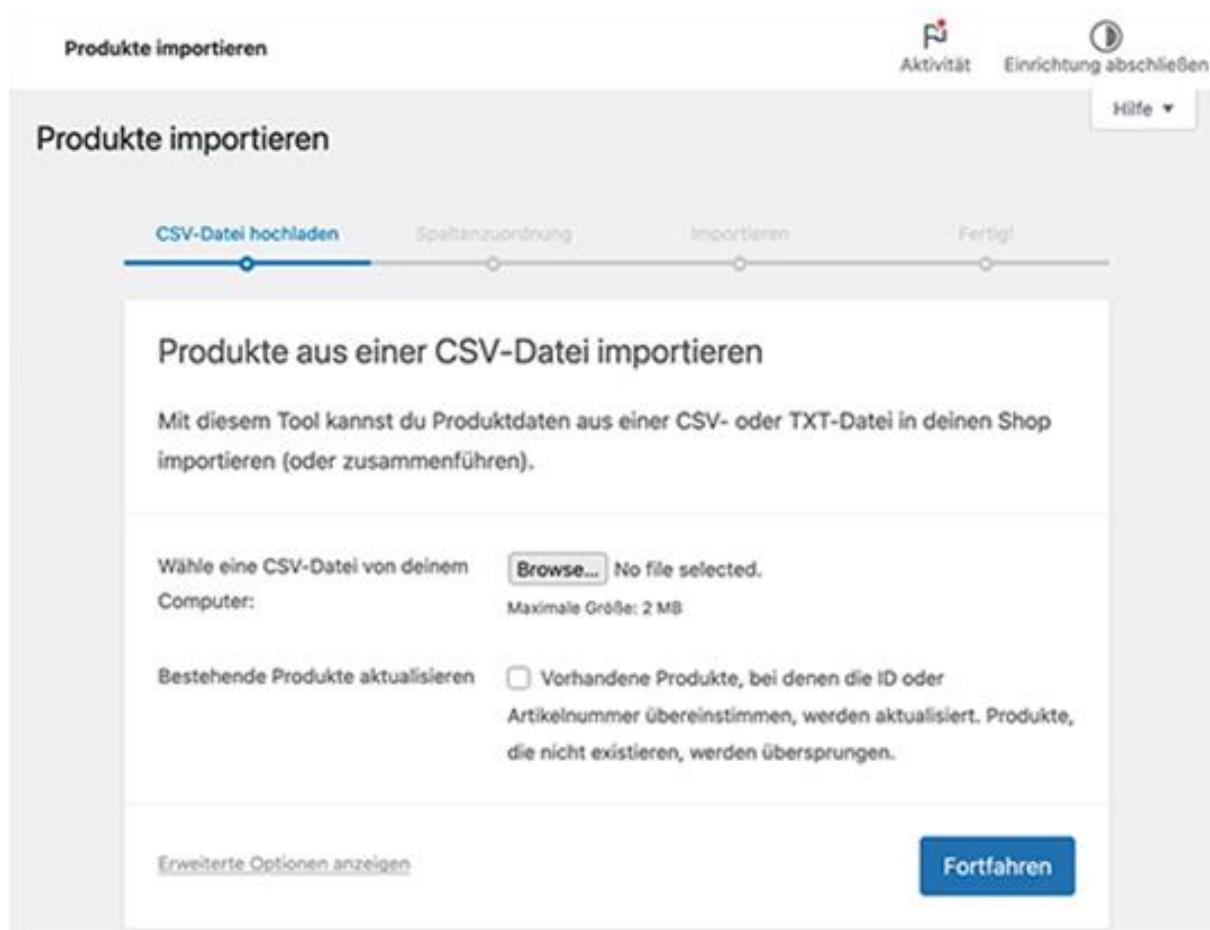


Abbildung 21: WooCommerce Import GUI 2

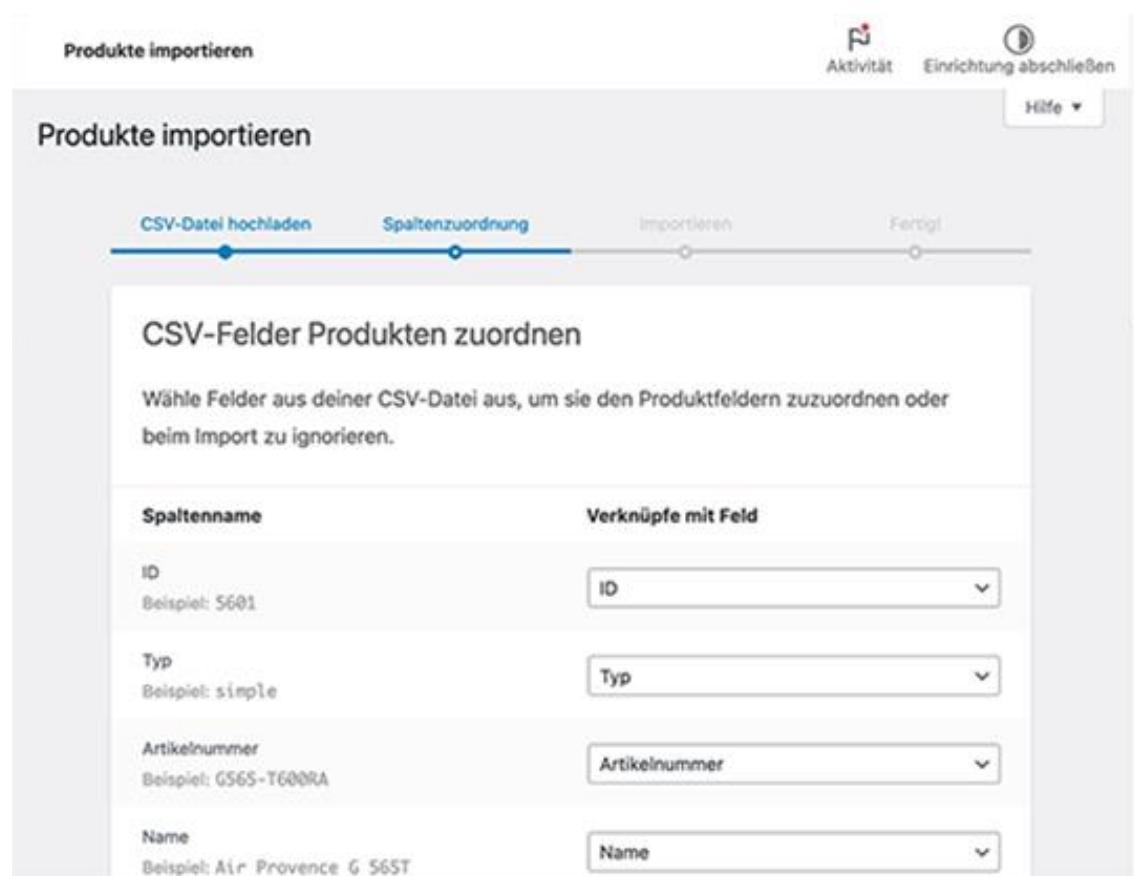


Abbildung 22: WooCommerce Import GUI 3

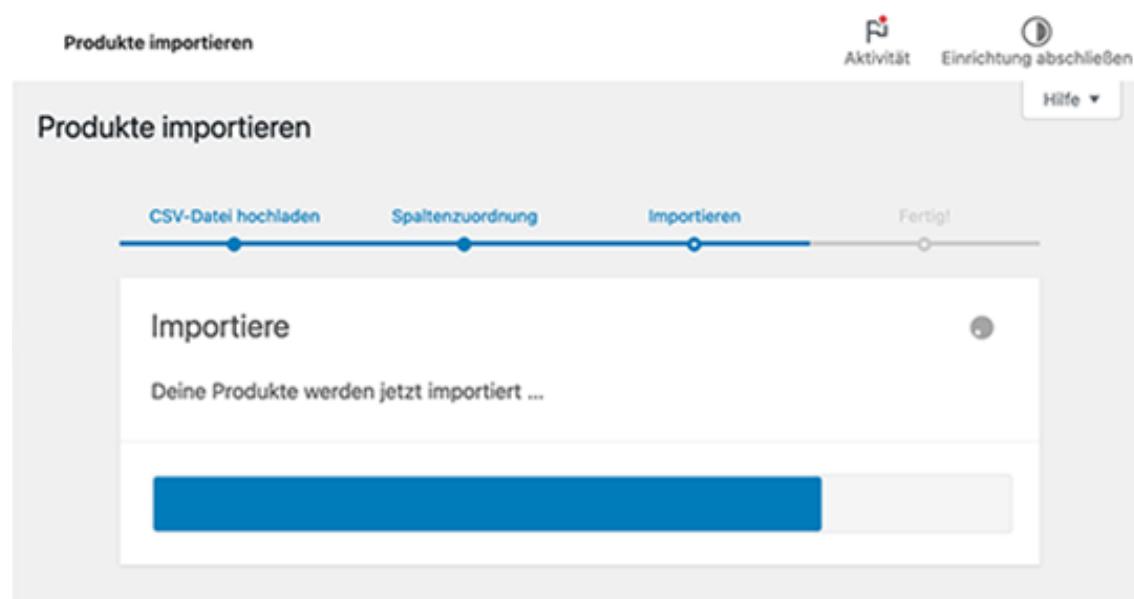


Abbildung 23: XLSX zu CSV

Abbildung 24: WooCommerce Fehler Meldung

