

cultus

webprojekt von jordan & timo

Inhaltsverzeichnis

# Projektbeschreib

## 1.1 Projektbeschreib

Mit unserem Projekt möchten wir einem Problem, dass die Welt schon lange hat, entgegenwirken. Der fehlende Komfort auf der Toilette. Jeder von uns weiss, wie unangenehm es sein kann, auf einem Thron zu sitzen, der einfach nicht passt. Deshalb haben wir eine Lösung entwickelt: ein Webshop, auf dem man Toiletten aus aller Welt kaufen kann. Ob dünn oder breit, ob bunt oder weiss. Wir haben alles! Mit unserer stolzen Auswahl von über 500 Exemplaren haben wir die besten Toilettentechnologien aus aller Welt gesammelt und verkaufen weltweit. Ungemütliche Geschäfte waren gestern!

## 1.2 Zieldefinierung

Als grundsätzliches Ziel haben wir einen Webshop. Folgende Funktionen sollte dieser haben:

* Benutzerverwaltung
* Ergonomisches Shop Design
* Datenbank für Benutzer und Produkte
* Zwei-Faktor-Authentifizierung
* Session Handling

## 1.3 Meilensteine

* Benutzer werden in der DB gespeichert und können bearbeitet werden
* Ein Benutzer kann sich einloggen
* Shop Page fertig (UI)
* Login Page fertig (UI)
* Produkte in DB gespeichert
* Dokumentation fertig
* Projekt fertig

# 2. Architektur

## 2.1 Toolwahl

Für die Realisierung des Tools haben wir uns für Technologien / Frameworks entschieden, die gut aufeinander abgestimmt sind. Für das Backend setzten wir auf nodejs / expressjs, da wir bereits damit Erfahrung gemacht haben und express sich bewährt hat. Für das Speichern von Daten verwendeten wir MongoDB, da es einer der einfachsten Wege ist SQL-Injections zu verhindern, da MongoDB eine NoSQL DB ist.

Nuxtjs, ein Framework das auf Vue, basiert war unsere Wahl fürs Frontend. Es ist ein simples, zuverlässiges Framework das das Darstellen von dynamischen Daten (z.B. Shop – Items) extrem einfach macht, was mit nur HTML/CSS/JS nicht so einfach realisierbar ist.

## 2.2 CRUD / Datenbankanbindung

Für den Zugriff auf die Daten verwenden wir wie schon erwähnt expressjs. Mit expressjs ist es möglich, API’s und Backends einfach und übersichtlich zu erstellen. Auch gibt es bereits bestehende Libraries für den MongoDB – Zugriff, was uns die Arbeit nochmal einfacher macht.

# 3. Arbeitsjournale

# 4. Datensicherheit

## 4.1 Zwei-Faktor-Authentifizierung

*2FA ist eine hervorragende Technologie, um die meisten Angriffsmethoden zu verhindern.*

Multi-Faktor-Authentifizierung (MFA; Multi-Faktor Authentifizierung oder 2FA, neben den anderen Begriffen) ist ein elektronisches Authentifizierungssystem, bei dem ein Benutzer nur dann Zugang zu einer Internetseite oder Anwendung erhält, wenn er einem Authentifizierungsmechanismus zwei oder mehr "Beweise" (oder Faktoren) vorlegt: Wissen (etwas, das nur der Benutzer weiß), Besitz (etwas, das nur der Benutzer hat) und Inhärenz (etwas, das nur der Benutzer ist). MFA bewahrt Benutzerdaten - zu denen private Identifikationsdaten oder finanzielle Vermögenswerte gehören können - davor, dass ein unbefugter Dritter, der z. B. ein einzelnes Passwort herausfinden könnte, darauf Zugriff hat.

Oft wird dies über eine dritt Anbieter App geregelt. Es wäre aber auch möglich über beispielsweise SMS oder E-Mail. Wir haben uns für letzteres entschieden. Wenn ein User sich einloggen möchte, generieren wir ein Token, dass aus Zufälligen Zeichen besteht und schicken dies an die vom Nutzer hinterlegte E-Mail Adresse. Dieses Token muss der Nutzer dann einfügen und erst wenn es übereinstimmt, kann der Nutzer sich einloggen.