

Basisdaten

Name: Raphael Fischer

Geb.: 06.09.1982

Adresse: Im Springen 10 | 42287 Wuppertal

Web: www.raphael-fischer.de

E-Mail: post@raphael-fischer.de

Mobil: 0176 / 320 45 346

Xing: [xing.com/profile/Raphael_Fischer5](https://www.xing.com/profile/Raphael_Fischer5)

LinkedIn: [linkedin.com/profile/view?id=112257950](https://www.linkedin.com/profile/view?id=112257950)



Lebenslauf

1989 – 2002: Schule Wuppertal | Abschluss *Abitur*

2002 – 2003: Praktikum 3D-Operator

2003 – 2005: Teilselbstständige Arbeit 3D-Operator

2005 – 2007: Studium *Sicherheitstechnik* | Wuppertal

Seit 2006: Selbstständig

2007 – 2009: Studium *Informatik* | Wuppertal

Seit 2009: Inhaber pro ci

2009 – 2010: Studium *Wirtschaftsinformatik* | FOM-

Düsseldorf

Top 5 Skills

- Spitzen Kaffee
- Schnelle Auffassungsgabe
- Konzeption und Umsetzung von systemübergreifenden Lösungen
- Javascript
- PHP

Skills

Consulting

- Beratung
- Projektmanagement
- Konzeption

Datenbanken

- SQL / MySQL
- MongoDB
- IndexedDB

- WebSQL
- Doctrine
- Sequelize

Programmier- /Scriptsprachen

- Javascript (ES6+) / Actionscript (Flash)
- PHP
- Bash (Shell)
- Dart

Frameworks / Bibliotheken

- Javascript
 - jQuery
 - Gulp / Grunt
 - Bower / NPM / Yarn
- CSS
 - Bootstrap
 - CoreUI
 - Sass
 - TailwindCSS
- PHP
 - Laravel 8 / 9
 - Voyager
 - Symfony
 - Yii 1 / Yii 2
 - Blade
 - Composer
- Mobil
 - Cordova / Phonegap
 - Flutter

Server / Virtualisierung

- Apache / NginX
- Ubuntu Server
- KVM
- ZFS
- Docker
- Node.js
- PHP / 5.4 – 8.x
- Git
- Gitlab
- RabbitMQ

- AWS Cloud (EC2, S3, CloudFront, ElastiCache, RDS, Route53)
- Zigbee2Mqtt
- Home Assistant
- OPNSense
- Proxmox

Technologien / Verfahren

- Ajax
- Git
- REST

Redaktionssysteme / CMS

- Wordpress
 - Template / Erweiterungen / Plugins
- Contao
 - Template / Erweiterungen / Plugins

Hardware / Ansteuerung

- Raspberry Pi
- Arduino
- Zigbee

Software / OS

- Windows
- Linux / Debian / Raspian / Ubuntu Server
- Arduino IDE
- OSX
- Photoshop / Illustrator
- Visual Studio Code
- Gitlab
- Github

Layout / Design

- Konzeption
- Design
- Photoshop / Illustrator / InDesign
- Fotografie Portrait / Landschaft / Makro / Industrial
- Lightroom

Referenzen

Projekt: Performance Optimierung des Online-Shop eines Startups, für einen TV-Auftritt.

Beschreibung: Dem Onlineshop des Startups „Matchachin“ sollten Beine gemacht werden.

Das Startup produziert einen Tee und ein Erfrischungsgetränk auf Basis von Guayusa, einer Pflanze aus dem Regenwald Equadors.

Für den Auftritt bei der Sendung „die beste Idee Deutschlands auf Vox, sollte der Online Shop an Performance zulegen.

Dies wurde durch den Umzug in die AWS-Cloud realisiert. In dieser lässt sich der Shop nun flexibel Skalieren.

Feature:

- Migration des Shopware Shops in die AWS Cloud
- CloudFlare als DNS
- Fallbacklösung auf archive.org
- CloudFront in Kombination mit S3-Bucket für eine schnelle Auslieferung der Assets.
- EC2 Instanzen mit skalierbarer CPU & Ram
- Weitere Staging VM für die Weiterentwicklung
- Auslagerung der Datenbank in Amazon RDS
- Auslagerung des Cache nach ElastiCache / Redis
- HA optimierte NginX Config
- HA optimierte PHP-FPM Config

Firma: <https://www.matchachin.com>

Entstehung: 2022

Technologien: AWS – EC2, S3, CloudFront, ElastiCache / Redis, RDS, CloudFlare, Shopware, NginX.



Projekt: Redesign Webseite.

Beschreibung: Layout, Konzeption, Fotografie, Texte und Umsetzung der Webseite der Tischlerei Knoche aus Wülfrath.

Feature:

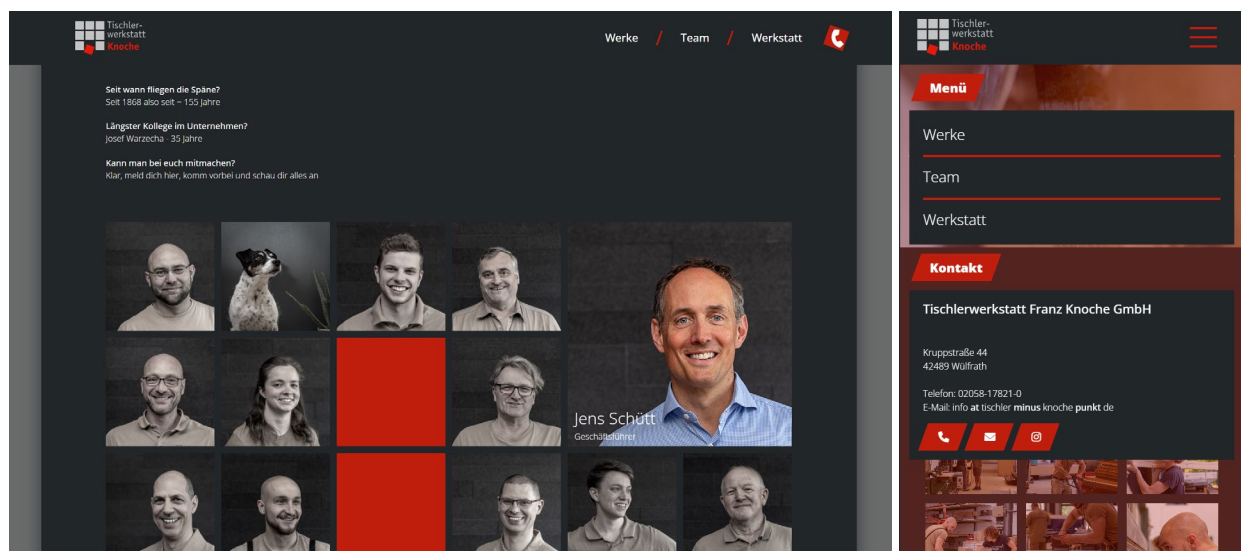
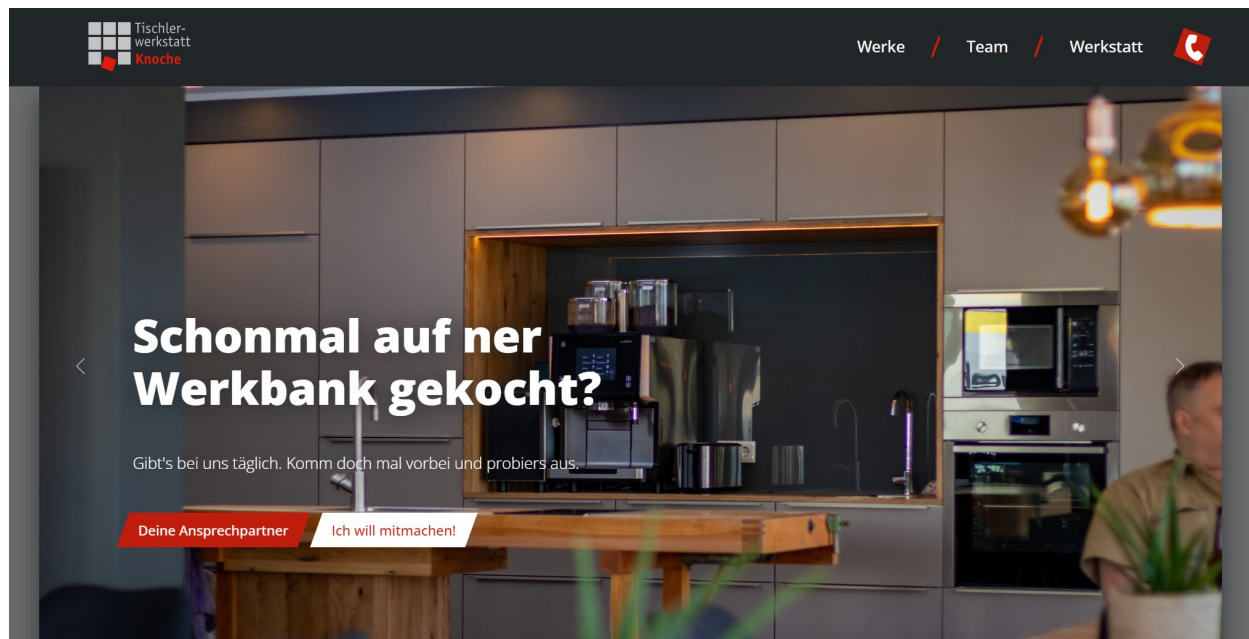
- Parallax Scrolling
- Team Galerie

- Werkstatt Galerie
- Referenzen
- Responsiv
- CSS-Animationen

url: <https://www.tischler-knoche.de>

Entstehung: 2022

Technologien: Js, jQuery, CSS, Ajax, PHP, Laravel, Blade.



Projekt: Programm für die Verarbeitung von Stitching Bildern (Windows)

Beschreibung: NodeJS Software zur automatischen Verarbeitung mehrerer Stitching Bildern auf Basis einer Konfiguration. Die Bilder werden unmittelbar nach ihrem „eintreffen“ (Thetering) verarbeitet und können mittels Flutter GUI überwacht werden. Nach der Verarbeitung werden die Bilder automatisch in Lightroom geladen.

Die GUI ermöglicht es darüber hinaus, die Konfiguration Live und vollumfänglich zu editieren.

Feature:

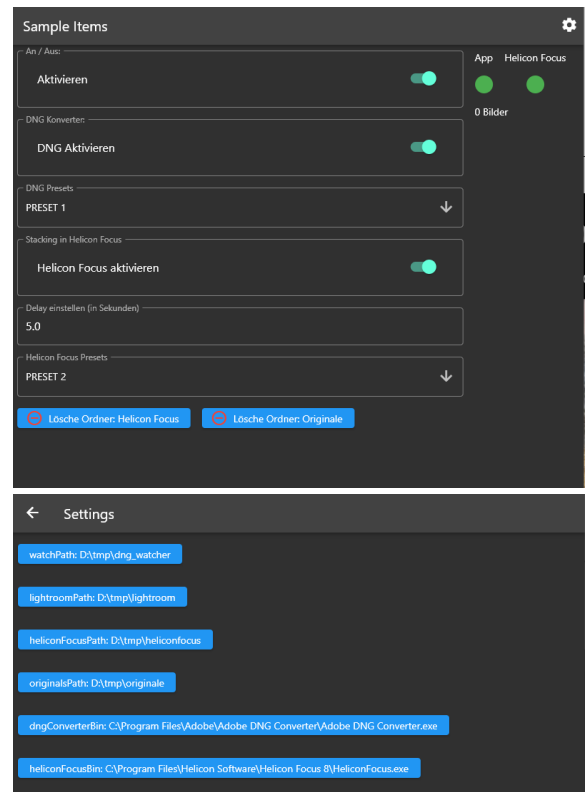
- Keine Datenbank!
- Einstellbares Delay der Bilder (Thetering)
- Einstellbare Windows Ordner per Ordner Picker
- Löschen von TMP Ordner per Button
- Anzeige über den Zustand der Software

- Automatisches Starten der NodeJS Scripte durch Flutter GUI

Firma: auf Anfrage

Entstehung: 2022

Technologien: Flutter, Dart, DNG Koverter, NodeJS, Git, Gitlab.



Projekt: Umsetzung Automatisierung für Fotografen (Flutter Windows)

Beschreibung: Vollautomatischer Service zur Vorbereitung, Katalogisierung, Verarbeitung von Bildern / Jobs in und außerhalb von Photoshop, inklusive Dashboard.

Auf Basis von NodeJS werden Bilder durch verschiedene Prozesse geleitet und per

Flutter GUI überwacht. Jedes Bild genießt hierbei eine völlig eigene Verarbeitung, basierend auf der individuellen Konfiguration des Bildes.

Darüber hinaus bietet das GUI einige Einstell-Möglichkeiten, welche den Workflow der Automatisierung beeinflussen können.

Das System arbeitet auf einem Server, (NodeJS) an dem verschiedene Clients (Flutter GUI) andocken können.

Feature:

- Auslesen von EXIF-Informationen
- Individuelle Automatisierung pro Bild für PS
- Steuerung der Node-Scripte über die GUI – AutoRestart / Konfiguration
- Multi Arbeitsplatz Version der GUI
- Filter
- Steuerung von Windows Prozessen

Firma: auf Anfrage

Entstehung: 2021 / 2022

Technologien: Flutter, Dart, NodeJS, Batch, Watcher, FTP Watcher, MySQL, Git, Gitlab, Photoshop, Javascript.

The screenshot shows the 'Task Overview' interface of the app. It features a table with columns: Fehler, Jobname, Bilder, Unterversion, Hochgeladen, Teilweise, Fertig, angelegt, and Letzte Änderung. The table lists tasks for 'Demo Kunde A' and 'Demo Kunde E'. Below the table, there is a 'Dashboard' button and a 'Job' button. The detailed view for 'Demo Kunde A' shows job statistics: Jobname: Demo Kunde A, Gesamt Anzahl: 14, Hauptversion: 13, angelegt: 2022-02-14 | 13:35:55, Hochgeladen: 2, Unterversion: 1, Letzte Änderung: 2022-03-30 | 11:16:36, Teilweise: 0, Status: uploaded, Fertig: 12. Below this, there is a table of file uploads with columns: Dateiname, Erstellt am, Zuletzt aktualisiert, and Status.

Fehler	Jobname	Bilder	Unterversion	Hochgeladen	Teilweise	Fertig	angelegt	Letzte Änderung	
	Demo Kunde A	13	1	2	0	12	2022-02-14 13:35:55	2022-03-30 11:16:36	<input type="checkbox"/>
	Demo Kunde E	3	0	0	0	3	2022-02-18 10:14:51	2022-03-28 11:44:46	<input type="checkbox"/>

Jobname	Gesamt Anzahl	Hauptversion	angelegt	Hochgeladen	Unterversion	Letzte Änderung	Teilweise	Status	Fertig
Demo Kunde A	14	13	2022-02-14 13:35:55	2	1	2022-03-30 11:16:36	0	uploaded	12

Dateiname	Erstellt am	Zuletzt aktualisiert	Status
2021-11-22cIMG_01403.jpg	2022-02-14 15:08:10	2022-02-14 15:08:35	99
2021-11-22cIMG_01401.jpg	2022-02-14 15:08:12	2022-03-30 11:16:13	99
2021-11-22cIMG_01411.jpg	2022-02-14 15:13:50	2022-03-29 14:31:16	99
211129a22885.jpg	2022-02-14 15:13:51	2022-03-29 14:31:16	99
2021-11-22cIMG_01449.jpg	2022-02-14 15:13:53	2022-03-24 21:05:12	20
211129a22887.jpg	2022-02-14 15:13:54	2022-03-28 12:12:17	99
2021-11-22cIMG_01407.jpg	2022-02-14 15:13:56	2022-03-28 12:12:19	99
211129a22881.jpg	2022-02-14 15:13:57	2022-03-29 14:31:16	99
211129a22878.jpg	2022-02-14 15:13:59	2022-03-29 14:31:16	99
2021-11-22cIMG_01415.jpg	2022-02-14 15:14:00	2022-02-14 15:18:25	99
2021-11-22cIMG_01441.jpg	2022-02-14 15:14:02	2022-03-24 21:05:53	20
2021-11-22cIMG_01434.jpg	2022-02-14 15:14:04	2022-02-14 15:20:12	99
211129a22883.jpg	2022-02-14 15:14:11	2022-03-28 12:08:19	99
2021-11-22cIMG_01401_v0-1.jpg	2022-03-29 15:06:30	2022-03-30 11:16:36	99

Projekt: Umsetzung App (iOS/Android/Desktop)

Beschreibung: Umsetzung einer App zur dezentralen Taskverwaltung für den internen Gebrauch. Ziel ist es, sich so weit wie möglich von externen Lösungen wie bspw. Google Firebase zu lösen, bei selben / ähnlichem Feature Umfang.

Dies wird realisiert durch das Zusammenspiel von Flutter und Appwrite.

Feature:

- Task erstellen / editieren / löschen
- Kategorien erstellen / editieren / löschen
- Bild aufnehmen und abspeichern
- Kalendereintrag erstellen / editieren / löschen

Firma: Eigenes Projekt

Entstehung: 2021 / noch in Arbeit

Technologien: Flutter, Dart, Appwrite, Docker, VM, Ubuntu Server, Git, Gitlab.

Projekt: Umsetzung App (iOS/Android)

Beschreibung: Umsetzung einer App zur Darstellung einer Wordpress-Webseite. Die Schnittstelle hierfür stellt eine Firebase Datenbank welche sowohl Artikel als auch Menüstruktur bereithält.

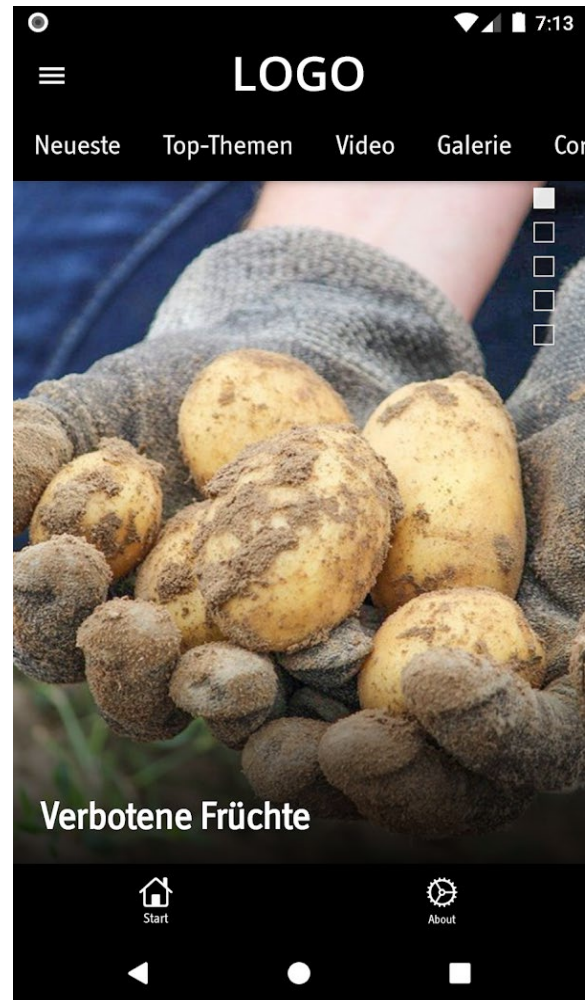
Feature:

- Daten-Anbindung über Firebase
- Dynamischer Menüaufbau
- Slide-Show vertikal und horizontal parallel
- Offlineverfügbarkeit
- Gitlab Auto-deploy

Firma: auf Anfrage

Entstehung: 2020 / 2021

Technologien: Flutter, Dart, Firebase, Git, Gitlab.



Projekt: Umsetzung App (iOS/Android)

Beschreibung: Konzeption und Umsetzung einer App (iOS/Android) für eine Studie des Fachbereichs Psychologie der Universität Bielefeld.

Feature:

- Authentifizierung
- API-Anbindung über REST / JSON
- Auto-Update per CodePush
- Modularer Aufbau
- Offline Verfügbarkeit
- Planbare Push-Notification
- Redux

Firma: Universität Bielefeld

Entstehung: 2019 – noch in Arbeit

Technologien: React Native, Redux, REST, JSON, Zapier, Google-Calendar, One-Signal.



Projekt: PoC / Konzeption

Beschreibung: Konzeption und Proof of Concept (PoC) einer Video on Demand Lösung (VoD), bestehend aus einer Verwaltungsebene (Backend) und einer Client Ebene (App - iOS / Android) für ein Startup aus Darmstadt.

Feature:

- Authentifizierung
- Such-Algorithmus
- API-Anbindung über REST / JSON
- Modularer Aufbau
- Streaming VoD

- DRM
- Offline Verfügbarkeit
- Auto-Update per CodePush
- Analytics Integration
- Verwaltung der Clients
- Verwaltung der Medien
- Bezahlungssystem

Firma: auf Anfrage

Entstehung: 2018 / 2019 – noch in Arbeit

Technologien: PHP, JS, CSS, GIT, Laravel, Wordpress, Woocommerce, React Native, REST, JSON.

Projekt: Redesign Web Portal.

Beschreibung: Übernahme der Entwicklungsabteilung, sowie Redesign und Re-Entwicklung eines Portals für virtuelle Messen (Expo-IP) www.expo-ip.com.

Feature:

- Neugestaltung der Oberfläche mit Schwerpunkt Responsive, sowie
- Neu Aufbau der gesamten Messe Plattform – Backend/Frontend
- Implementierung diverser API's für:
 - Registrierung / Einbindung von GoToWebinar Webinaren
 - Registrierung von Usern per Webhook über Drittanbieter

- Manuell / Zeitgesteuerte Pop-Up's
- Implementierung diverser Widget's zur individuellen Gestaltung der Messeplattform
- iCal Generierung.
- Dedizierte Messen / Whitelabel - Lösungen durch Docker
- Uvm.

Firma: Expo-IP

Entstehung: 2018 - 2021

Technologien: PHP, JS, jQuery, CSS, HTML, Ajax, Yii2-Framework, REST, JSON, Material Design, Gulp, GIT, GitLab.

.

Projekt: PoC App

Beschreibung: Proof of Concept einer öffentlichen App (iOS / Android) für ein mittelständisches Unternehmen in Düsseldorf.

Feature:

- Authentifizierung

- Such-Algorithmus
- API-Anbindung über REST / JSON
- Modularer Aufbau
- Auto-Update per CodePush
- Analytics Integration

Firma: auf Anfrage

Entstehung: 2018

Technologien: JS, React Native, REST, JSON.

Projekt: IoT Prototyp App / Hardware / Cloud.

Beschreibung: Prototyp, für ein offenes System, zur dezentrale Erfassung, Verarbeitung und Steuerung von Daten /

Aktoren, mit Kontrolleinheit via Smartphone (iOS / Android).

Feature:

- Zeit und Ereignisgesteuerte Aktoren

- Gesamtkontrolle via Dashboard (Cloud)
- Partielle Steuerung durch App
- Modularisierte Bauweise der Aktoren

App:

- Direktverbindung mit dem Host via WLAN
- Authentifizierung am Host
- Ausgabe von Werten
- Eingabe von Werten
- Steuerung von Aktoren

Host (Raspberry Pi):

- Direktverbindung mit dem Client via WLAN
- Direktverbindung zu den Aktoren (Arduino) via 2400Ghz Radio / Funk.
- Daten Synchronisierung in die Cloud
- Daten Ein / Ausgabe
- Datenspeicherung
- Datenweiterverarbeitung

- Verwaltung mehrerer Aktoren
- Ereignisverarbeitung

Aktor (Arduino pro mini):

- Auslesen von Daten div. Sensoren
- Steuern div. Aktoren
- Autark - Batterie betrieben
- Kompakte Bauweise.

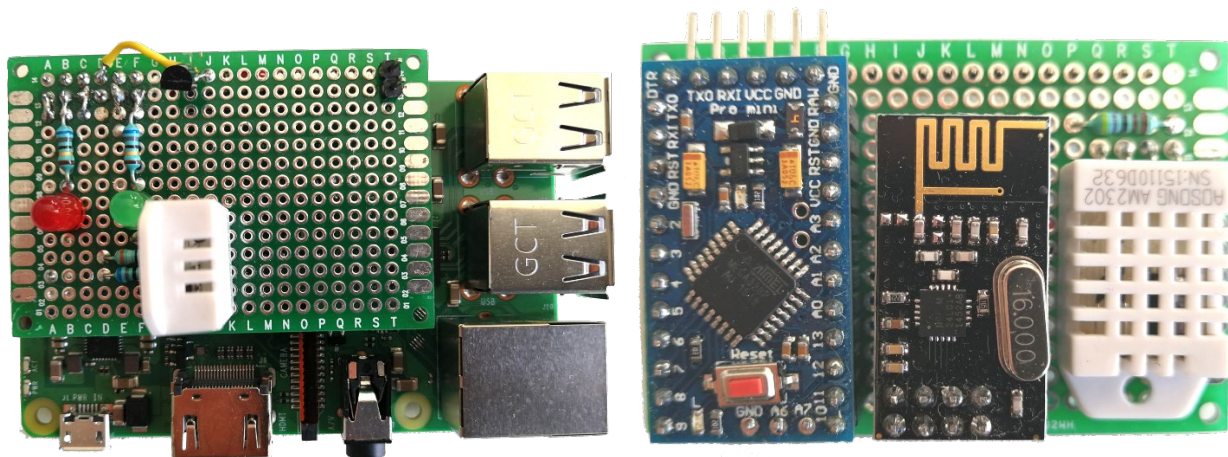
Gehäuse für Aktor / Host (in Arbeit):

- 3D Model aus Blender
- Optisch ansprechend
- Für den jeweiligen Aktor und Kontext angepasst
- Druckmaterial: PLA

Firma: eigenes Projekt

Entstehung: 2018, noch in Entwicklung

Technologien / Software: React Native, Node.js, iOS, Android, Raspberry Pi, Arduino, PHP, CSS, SCSS, C++, REST, JSON, Ajax, Photoshop, Blender (3D), Cura 3D, VS Code.





Projekt: Redesign / Umsetzung hybride App.

Beschreibung: Umsetzung einer öffentlichen App (iOS / Android) für ein mittelständisches Unternehmen in Düsseldorf.

Feature:

- Authentifizierung

- Such-Algorithmus
- API-Anbindung über REST / JSON
- Modularer Aufbau
- Auto-Update per CodePush
- Analytics Integration

Firma: auf Anfrage

Entstehung: 2017, noch in Entwicklung

Technologien: ExtJS, CSS, SCSS, Sencha Architect, Phonegap, REST, JSON.

Projekt: Design / Umsetzung interne App.

Beschreibung: Umsetzung mehrerer interner App's (iOS / Android) für ein mittelständisches Unternehmen in Düsseldorf.

Feature:

- Authentifizierung
- Offline Verfügbar
- Projektverwaltung
- Kameranutzung

- Barcode-Scanner
- API-Anbindung über REST / JSON
- Voice-Recorder
- Modularer Aufbau
- Multilingual
- Auto-Update
- Analytics Integration

Firma: auf Anfrage

Entstehung: 2016 / 2018, noch in Entwicklung

Technologien: ExtJs, CSS, SCSS, Sencha Architect, Phonegap, REST, JSON.

Projekt: Umsetzung Frontend.

Beschreibung: Umsetzung eines komplexen Frontend-Designs für den Ski- und Snowboard-Reiseveranstalter **E&P**.

Reisen im Auftrag der Agentur dasMinisterium.

Feature:

- Flipp-Cards
- Responsiv

- Responsive SVG-Grafiken
- Komplexe Navigation
- Multiple Farbschematas
- CSS-Animationen

url: www.ep-reisen.de

Entstehung: 2016

Technologien: Js, jQuery, CSS, Gulp, Git, PHP.

Projekt: Redesign Webseite.

Beschreibung: Gestaltung und Umsetzung der Webseite der Tischlerei Knoche aus Wuppertal.

Feature:

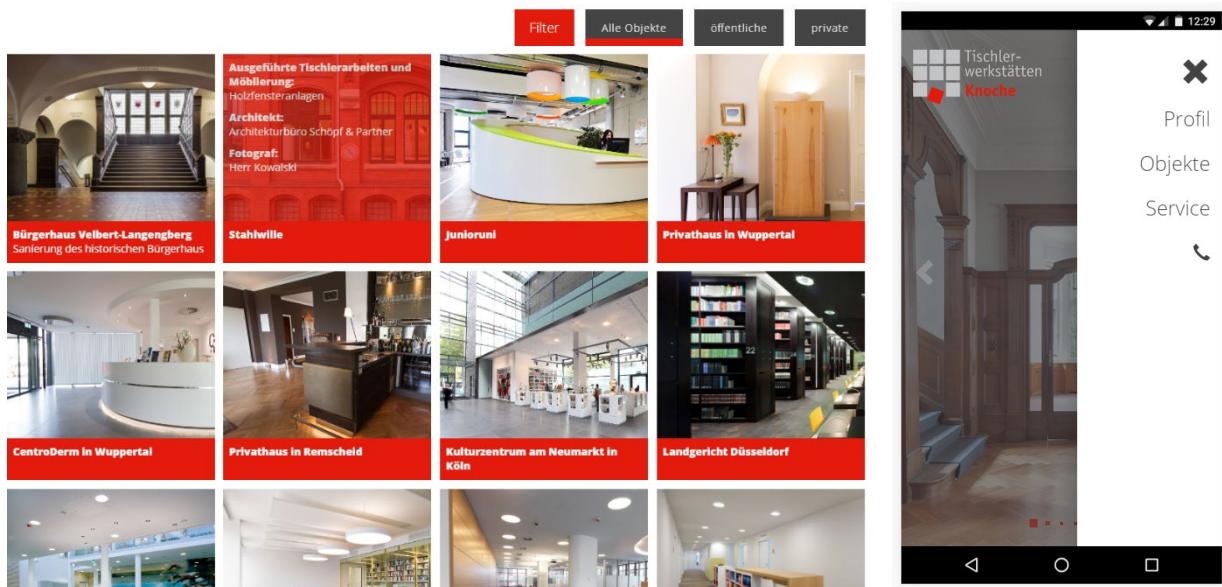
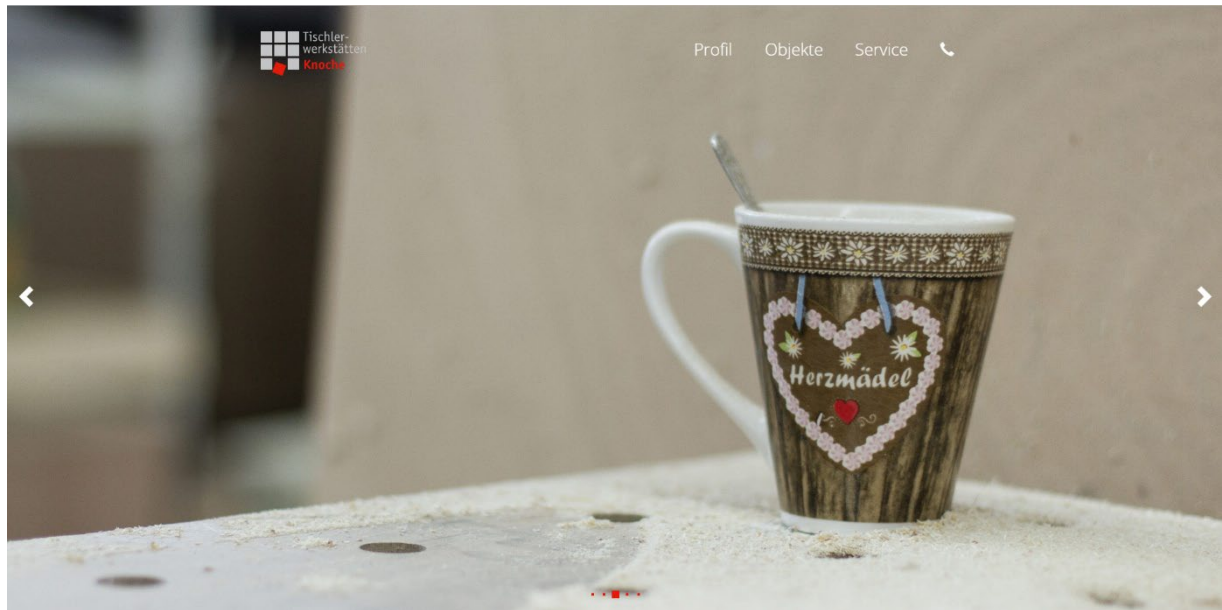
- Parallax Scrolling
- Referenzgalerie mit Ajax-Loader
- Responsiv

- Lazy loading
- CSS-Animationen

url: www.tischler-knoche.de

Entstehung: 2016

Technologien: Js, jQuery, CSS, Ajax, PHP, Contao.



Projekt: Umsetzung Buchungsformular.

Beschreibung: Umsetzung eines komplexen Buchungsformulars für die Surfschule **Maximum Surfcamp** im Auftrag der Agentur dasMinisterium.

Feature:

- Multilingual

- Kontextabhängige Auswahl

- Validierung

- Flexibel erweiterbar

url: www.maximumsurfcamp.com/buchen

Entstehung: 2016

Technologien: Js, jQuery, CSS, Gulp, Git, PHP.

Beschreibung: Entwicklung einer Web-App für die Integration des Teamy Buchungssystems in eine bestehende Webseite. Die Web-App hat folgende Feature:

- Einfaches Code-Snippet für die Einbindung in die bestehende Webseite
- Vollständiger Kalender
- 3 verschiedene Ansichten

- Vollständige Synchronisation mit dem internen Teamy-Kalender.
- Buchungen direkt über den Kalender durchführbar
- Buchungsübersicht im internen Kalender

url: www.teamy.eu

Entstehung: 2016

Technologien: Js, requireJs, gulp, jQuery, CSS, Ajax, PHP, Codeigniter, individuell angepasste Bootstrap Version.

Zurück

Dschungel-Party

Kindergeburtstag 5-11 Jahre

Termindetails

Datum:

23.02.2017

Uhrzeit:

16:00 - 18:00

Status:

Es sind noch 13 Tickets verfügbar

Buchungsdaten

Firma, Institut

Firma, Institut

Vorname

Vorname

Nachname

Nachname

Adresse

Adresse

PLZ, Ort

PLZ, Ort

Telefon

Telefon

E-Mail

E-Mail

Zahlungsort

Direkt vor Ort ▾

Nachricht

Preise

1-5 Kids :	135.00 €	<input type="text" value="0"/>
weiteres Kind:	10.00 €	<input type="text" value="0"/>
Hotdog-Party-Paket (Preis pro Kind):	5.50 €	<input type="text" value="0"/>
9-12 Kids(ab 01.03.17):	280.00 €	<input type="text" value="0"/>
1-8 Kids (ab 01.03.17):	190.00 €	<input type="text" value="0"/>
Summe:	0€	

☒ AGB gelesen und akzeptiert

Kostenpflichtig buchen

<

>

Heute

Februar 2017

MonatWocheTag

Mo.	Di.	Mi.	Do.	Fr.	Sa.	So.
30 16:00 Dschungel Party 16:00 Dschungel Party	31 16:00 Dschungel Party	1 16:00 Dschungel Party	2 16:00 Dschungel Party	3 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	4 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	5 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party
6 16:00 Dschungel Party 16:00 Dschungel Party	7 16:00 Dschungel Party	8 16:00 Dschungel Party	9 16:00 Dschungel Party	10 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	11 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	12 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party
13 16:00 Dschungel Party 16:00 Dschungel Party	14 16:00 Dschungel Party	15 16:00 Dschungel Party	16 16:00 Dschungel Party	17 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	18 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	19 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party
20 16:00 Dschungel Party 16:00 Dschungel Party	21 16:00 Dschungel Party	22 16:00 Dschungel Party	23 16:00 Dschungel Party	24 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	25 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	26 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party
27 16:00 Dschungel Party 16:00 Dschungel Party	28 16:00 Dschungel Party	1 16:00 Dschungel Party	2 16:00 Dschungel Party	3 16:00 Dschungel Party	4 16:00 Dschungel Party	5 16:00 Dschungel Party
6 16:00 Dschungel Party	7 16:00 Dschungel Party	8 16:00 Dschungel Party	9 16:00 Dschungel Party	10 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	11 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party	12 16:00 Dschungel Party 16:00 Dschungel Party 16:00 Dschungel Party

Information: Nach dem Absenden bekommen Sie eine E-Mail mit Ihren Buchungsdaten und einem Bestätigungslink zugesendet. Bitte klicken Sie auf den Link um Ihr Buchung zu bestätigen, erst dann wird sie verbindlich. Online-Buchungen sind bis zu 24 Stunden vor Kursbeginn möglich. Bitte beachten Sie, dass Kursausfälle auf unserer Facebook-Seite angekündigt werden. Für kurzfristige Buchungen rufen Sie uns bitte an: 0721 9880950

Projekt: Umsetzung Redesign Webseite.

Beschreibung: Umsetzung der Webseite der Agentur Ideenservice in Düsseldorf.
Feature:

- Parallax Scrolling
- Referenzgalerie mit Ajax-Loader
- Integration von Video / Audio

- Blätterbares Magazin.

url: www.ideenservice.de

Entstehung: 2016, auf Wunsch des Kunden noch kein responsives Design

Technologien: Js, jQuery, CSS, Ajax, PHP, Contao.

Projekt: Backend/Frontend für 3M

Beschreibung: Teil der Entwicklung für das Frontend/Backend für eine Interne/externe Software von 3M im Auftrag der Agentur Blanko.

Entstehung: 2016, noch in der Entwicklung

Technologien: Js, jQuery, CSS, Ajax, PHP, Laravel, PHPunit, Bootstrap.

Projekt: Umsetzung Redesign Webseite

Beschreibung: Die Umsetzung einer responsiven multidomain Webseite für die Agentur dasMinisterium. Der Clou: Neben individuell programmierten Modulen wird Google Analytics nicht nur von einer, sondern von allen angeschlossenen Domains erfasst. Somit lässt sich der gesamte Weg eines Nutzers verfolgen. Das CMS der Wahl war CONTAO. Neben den Basisfeature wie:

- Mobil-optimierung,
- SEO
- SEO – multidomain
- einfache Animationen
- Galerie

wurden auch individuellere Feature für den Kunden programmiert wie:

- Personalverwaltung
- Speisekartenverwaltung
- Newspool für die Hauptseite
- Hochzeitsplaner

url: www.culinaria-wuppertal.de

Entstehung: 2015

Technologien: Js, jQuery, CSS, Ajax, PHP, Contao, Modulprogrammierung für die Speisekarten und Personalverwaltung, sowie die Programmierung domainübergreifender Newssammlungen. Domainübergreifendes Google-Analytics.

Projekt: Micro-Reservierungssystem

Beschreibung: System zur Auswahl und Reservierung gewünschter Speisen, mit anschließendem Ausdruck bei dem Anbieter. Die Produkte lassen sich bequem über ein Wordpress-Interface einstellen.

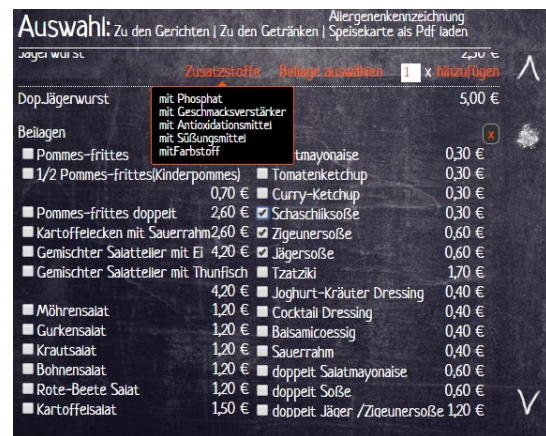
url: www.grillstube-unkel.de

Entstehung: 2012

Technologien: Js, jQuery, CSS, Ajax, PHP, Wordpress, Verwaltung der Produkte durch Wordpressintegration.



Menü-Übersicht



Detail Auswahl



Detail Bestellung

Projekt: Formularbuilder / Datenerfassung / Datenexport.

Beschreibung: Das Formulare System wurde entwickelt um die Papierlose-Datenerfassungen eines mittelständischen Unternehmens zu ermöglichen. Es besteht aus den folgenden Komponenten:

- Projektverwaltung
- Formular/Polleerstellung
- Dateneingabe - web
- Dateneingabe - mobil online/offline
- Polleingabe & export csv/mysql
- Datenexport nach Excel

Die Formularzuweisung geschieht über die Projektverwaltung; Die Formulareingabe kann über das Webinterface oder ein mobiles Endgerät online und offline erfolgen.

url: Interne Kundenapplikation

Entstehung: 2013

Technologien: JS, Ext-JS, jQuery, CSS, Ajax, indexedDB, mongoDB, MySQL, MVC, PHP 5, Yii-Framework, XPHPEXcel, Twitter Bootstrap.

Name	Erstellt am	Erstellt von	Bearbeitet am	Bearbeitet von		
Kaskade	Tue, 03 Sep 2013 12:58:58 +0200	admin	2013-09-11 13:19:28	admin		
Begehung-Test-SC-2013-09-12	Thu, 12 Sep 2013 15:40:24 +0200	admin	2013-10-14 09:24:49	admin		
testformular	Fri, 13 Sep 2013 13:37:03 +0200	admin	2013-09-13 14:12:18	admin		
testo	Fri, 13 Sep 2013 13:37:01 +0200	admin	2013-09-13 14:12:12	admin		
Umfrage1	Fri, 13 Sep 2013 14:13:29 +0200	admin	2014-02-13 18:36:50	admin		
test01	Mon, 16 Sep 2013 09:18:16 +0200	admin	2013-09-18 09:12:42	admin		
Begehung1C	Tue, 24 Sep 2013 16:57:07 +0200	admin	2013-09-24 17:03:51	admin		
test02	Wed, 04 Oct 2013 12:06:27 +0200	admin	2013-10-09 12:06:43	admin		
CheckUp	Fri, 10 Jan 2014 14:54:09 +0100	wilde	2014-01-10 15:25:39	wilde		
021	Wed, 15 Jan 2014 17:51:57 +0100	wilde	2014-01-20 16:12:11	wilde		

Formularübersicht

Formular editieren.
Name: testformular

Formular: testformular

id: Element:

test1: Textzeile

checkbox: ☐ Checkbox

test2: **test2** **test2**

test3: **test3** **test3**

test4: **test4** **test4**

test5: **test5** **test5**

test6: **test6** **test6**

test7: **test7** **test7**

test8: **test8** **test8**

test9: **test9** **test9**

test10: **test10** **test10**

test11: **test11** **test11**

test12: **test12** **test12**

test13: **test13** **test13**

test14: **test14** **test14**

test15: **test15** **test15**

test16: **test16** **test16**

test17: **test17** **test17**

test18: **test18** **test18**

test19: **test19** **test19**

test20: **test20** **test20**

test21: **test21** **test21**

test22: **test22** **test22**

test23: **test23** **test23**

test24: **test24** **test24**

test25: **test25** **test25**

test26: **test26** **test26**

test27: **test27** **test27**

test28: **test28** **test28**

test29: **test29** **test29**

test30: **test30** **test30**

test31: **test31** **test31**

test32: **test32** **test32**

test33: **test33** **test33**

test34: **test34** **test34**

test35: **test35** **test35**

test36: **test36** **test36**

test37: **test37** **test37**

test38: **test38** **test38**

test39: **test39** **test39**

test40: **test40** **test40**

test41: **test41** **test41**

test42: **test42** **test42**

test43: **test43** **test43**

test44: **test44** **test44**

test45: **test45** **test45**

test46: **test46** **test46**

test47: **test47** **test47**

test48: **test48** **test48**

test49: **test49** **test49**

test50: **test50** **test50**

test51: **test51** **test51**

test52: **test52** **test52**

test53: **test53** **test53**

test54: **test54** **test54**

test55: **test55** **test55**

test56: **test56** **test56**

test57: **test57** **test57**

test58: **test58** **test58**

test59: **test59** **test59**

test60: **test60** **test60**

test61: **test61** **test61**

test62: **test62** **test62**

test63: **test63** **test63**

test64: **test64** **test64**

test65: **test65** **test65**

test66: **test66** **test66**

test67: **test67** **test67**

test68: **test68** **test68**

test69: **test69** **test69**

test70: **test70** **test70**

test71: **test71** **test71**

test72: **test72** **test72**

test73: **test73** **test73**

test74: **test74** **test74**

test75: **test75** **test75**

test76: **test76** **test76**

test77: **test77** **test77**

test78: **test78** **test78**

test79: **test79** **test79**

test80: **test80** **test80**

test81: **test81** **test81**

test82: **test82** **test82**

test83: **test83** **test83**

test84: **test84** **test84**

test85: **test85** **test85**

test86: **test86** **test86**

test87: **test87** **test87**

test88: **test88** **test88**

test89: **test89** **test89**

test90: **test90** **test90**

test91: **test91** **test91**

test92: **test92** **test92**

test93: **test93** **test93**

test94: **test94** **test94**

test95: **test95** **test95**

test96: **test96** **test96**

test97: **test97** **test97**

test98: **test98** **test98**

test99: **test99** **test99**

test100: **test100** **test100**

test101: **test101** **test101**

test102: **test102** **test102**

test103: **test103** **test103**

test104: **test104** **test104**

test105: **test105** **test105**

test106: **test106** **test106**

test107: **test107** **test107**

test108: **test108** **test108**

test109: **test109** **test109**

test110: **test110** **test110**

test111: **test111** **test111**

test112: **test112** **test112**

test113: **test113** **test113**

test114: **test114** **test114**

test115: **test115** **test115**

test116: **test116** **test116**

test117: **test117** **test117**

test118: **test118** **test118**

test119: **test119** **test119**

test120: **test120** **test120**

test121: **test121** **test121**

test122: **test122** **test122**

test123: **test123** **test123**

test124: **test124** **test124**

test125: **test125** **test125**

test126: **test126** **test126**

test127: **test127** **test127**

test128: **test128** **test128**

test129: **test129** **test129**

test130: **test130** **test130**

test131: **test131** **test131**

test132: **test132** **test132**

test133: **test133** **test133**

test134: **test134** **test134**

test135: **test135** **test135**

test136: **test136** **test136**

test137: **test137** **test137**

test138: **test138** **test138**

test139: **test139** **test139**

test140: **test140** **test140**

test141: **test141** **test141**

test142: **test142** **test142**

test143: **test143** **test143**

test144: **test144** **test144**

test145: **test145** **test145**

test146: **test146** **test146**

test147: **test147** **test147**

test148: **test148** **test148**

test149: **test149** **test149**

test150: **test150** **test150**

test151: **test151** **test151**

test152: **test152** **test152**

test153: **test153** **test153**

test154: **test154** **test154**

test155: **test155** **test155**

test156: **test156** **test156**

test157: **test157** **test157**

test158: **test158** **test158**

test159: **test159** **test159**

test160: **test160** **test160**

test161: **test161** **test161**

test162: **test162** **test162**

test163: **test163** **test163**

test164: **test164** **test164**

test165: **test165** **test165**

test166: **test166** **test166**

test167: **test167** **test167**

test168: **test168** **test168**

test169: **test169** **test169**

test170: **test170** **test170**

test171: **test171** **test171**

test172: **test172** **test172**

test173: **test173** **test173**

test174: **test174** **test174**

test175: **test175** **test175**

test176: **test176** **test176**

test177: **test177** **test177**

test178: **test178** **test178**

test179: **test179** **test179**

test180: **test180** **test180**

test181: **test181** **test181**

test182: **test182** **test182**

test183: **test183** **test183**

test184: **test184** **test184**

test185: **test185** **test185**

test186: **test186** **test186**

test187: **test187** **test187**

test188: **test188** **test188**

test189: **test189** **test189**

test190: **test190** **test190**

test191: **test191** **test191**

test192: **test192** **test192**

test193: **test193** **test193**

test194: **test194** **test194**

test195: **test195** **test195**

test196: **test196** **test196**

test197: **test197** **test197**

test198: **test198** **test198**

test199: **test199** **test199**

test200: **test200** **test200**

test201: **test201** **test201**

test202: **test202** **test202**

test203: **test203** **test203**

test204: **test204** **test204**

test205: **test205** **test205**

test206: **test206** **test206**

test207: **test207** **test207**

test208: **test208** **test208**

test209: **test209** **test209**

test210: **test210** **test210**

test211: **test211** **test211**

test212: **test212** **test212**

test213: **test213** **test213**

test214: **test214** **test214**

test215: **test215** **test215**

test216: **test216** **test216**

test217: **test217** **test217**

test218: **test218** **test218**

test219: **test219** **test219**

test220: **test220** **test220**

test221: **test221** **test221**

test222: **test222** **test222**

test223: **test223** **test223**

test224: **test224** **test224**

test225: **test225** **test225**

test226: **test226** **test226**

test227: **test227** **test227**

test228: **test228** **test228**

test229: **test229** **test229**

test230: **test230** **test230**

test231: **test231** **test231**

test232: **test232** **test232**

test233: **test233** **test233**

test234: **test234** **test234**

test235: **test235** **test235**

test236: **test236** **test236**

test237: **test237** **test237**

test238: **test238** **test238**

test239: **test239** **test239**

test240: **test240** **test240**

test241: **test241** **test241**

test242: **test242** **test242**

test243: **test243** **test243**

test244: **test244** **test244**

test245: **test245** **test245**

test246: **test246** **test246**

test247: **test247** **test247**

test248: **test248** **test248**

test249: **test249** **test249**

test250: **test250** **test250**

test251: **test251** **test251**

test252: **test252** **test252**

test253: **test253** **test253**

test254: **test254** **test254**

test255: **test255** **test255**

test256: **test256** **test256**

test257: **test257** **test257**

test258: **test258** **test258**

test259: **test259** **test259**

test260: **test260** **test260**

test261: **test261** **test261**

test262: **test262** **test262**

test263: **test263** **test263**

test264: **test264** **test264**

test265: **test265** **test265**

test266: **test266** **test266**

test267: **test267** **test267**

test268: **test268** **test268**

test269: **test269** **test269**

test270: **test270** **test270**



Auswahl des Stationstyp

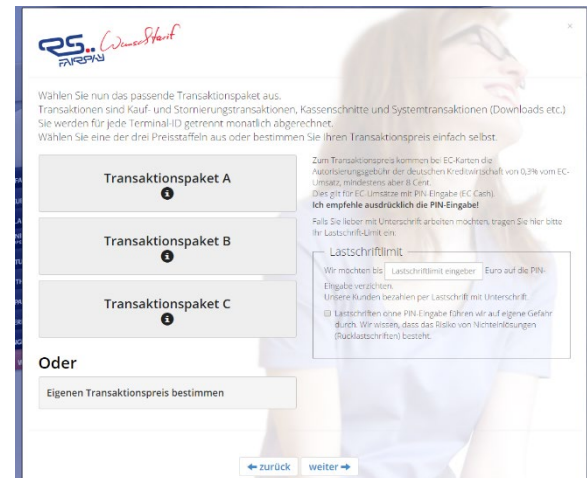


Projekt: Gymnasium Alsdorf - Redesign
Beschreibung: Redesign des Webauftritt des Gymnasium Alsdorf auf Basis des CMS Contao. Diverse Feature wie:

- Anlage von Lehrer
- Umfangreiches Menü auf Basis der Artikelstruktur



Auswahl der Basisstation mit Preisvorstellung



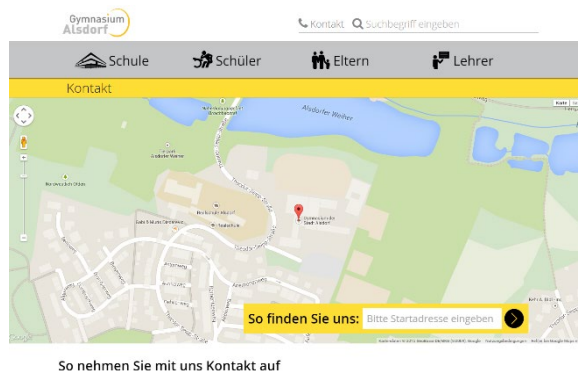
Auswahl eines Transaktionstyp

- Kontakt mit Google-Maps Routenplaner
 - Schnittstelle zu Smartsheet für die Kalenderdarstellung
- url:** www.daltongymnasium-alsdorf.de
Entstehung: 2015
Technologien: Js, jQuery, CSS, Ajax, Bootstrap, PHP, Contao, REST über Curl.

Startseite



Menü



Kontaktseite – Google Maps

Name	E-Mail	Raum	Elternsprechtag	Fach 1	Fach 2	Fach 3
Adam, Charlotte	Adam@gymnasium-alsdorf.de	N13		Latein	Geschichte	
Altintas, Sena	altintas@gymnasium-alsdorf.de	S14		Mathematik	Spanisch	
Anber, Trang	Anber@gymnasium-alsdorf.de	S 23		Englisch	Französisch	
Appelhans, Maik	Appelhans@gymnasium-alsdorf.de Termin vereinbaren	Q31		Mathematik	Physik	
Berlin, Aenna	berlin@gymnasium-alsdorf.de	Q48		Geographie	Chemie	Physik
Bock, Wilfried	Schulleiter@gymnasium-alsdorf.de Termin vereinbaren	N23	Elternsprechtag	Mathematik	Geographie	
Braun, Anja	Braun@gymnasium-alsdorf.de Termin vereinbaren	N12		Mathematik	Biologie	
Brosde, Edda	Brosde@gymnasium-alsdorf.de	N24		Mathematik	Französisch	
Dohmer, Eva	Dohmer@gymnasium-alsdorf.de Termin vereinbaren	Q31		Mathematik	Physik	
Dr. Ebel, Matthias	Ebel@gymnasium-alsdorf.de Termin vereinbaren	Q45		Biologie	Chemie	

Kollegium mit integrierter Suche.

Projekt: App Entwicklung - aktiv

Beschreibung: Entwicklung einer App mit folgenden Feature:

- Erfassen von Daten
- Erinnerungsfunktion mit lokalem Push
- Berechnung einer Statistik mit Tagesgenauem Durchschnitt auf Basis eingegebener Daten.
- Login App / Server
- Restschnittstelle zu Login und Datenabgleich.

- Modulentwicklung mit AMD
- Synchronisation wenn online.
- Erweiterung vorhandener Module.

url: noch in der Entwicklung, untersteht der Schweigepflicht.

Entstehung: 2014-2015 noch aktiv

Technologien: Js, jQuery, CSS, Ajax, PHP, MySQL, WebSQL, Cordova/Phonegap, Intel App Framework, REST über PHP/Ajax, JSON, requireJS, AMD.