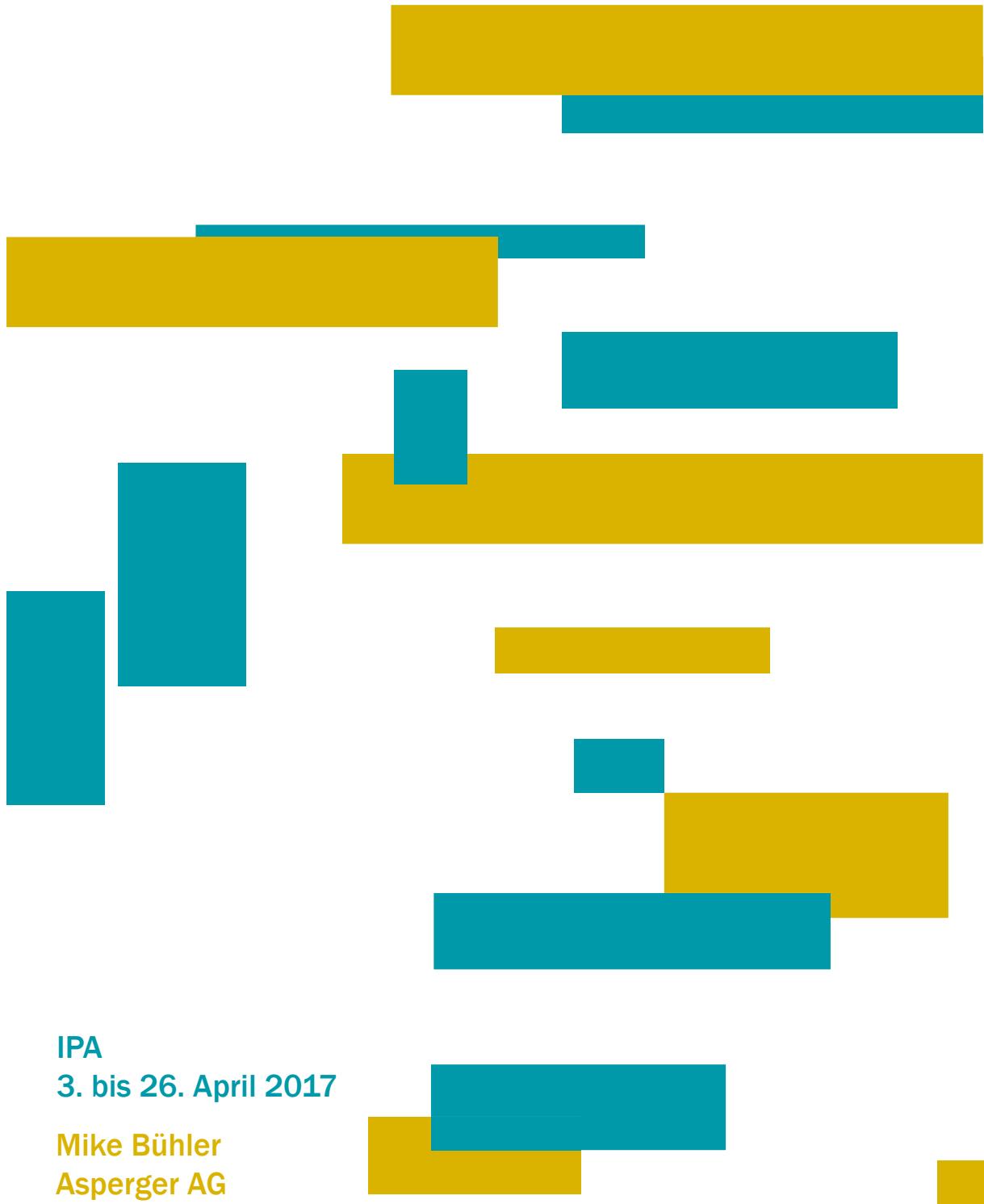


# Dokumentation Formulargenerator



IPA  
3. bis 26. April 2017

Mike Bühler  
Asperger AG

## Teil 1: IPA-Umfeld und -Ablauf

Aufgabenstellung	4
Projektorganisation	6
Vorkenntnisse	6
Vorarbeiten	7
Firmenstandards	7
Zeitplan	8
Arbeitsprotokoll	9
Organisation der Arbeitsergebnisse	15
Wirtschaftlichkeitsanalyse	15
Zielpublikumsanalyse	16

## Teil 2: Projektdokumentation

Management-Summary	18
Ausgangslage	18
Vorgehen	18
Ergebnis	19
Formulargenerator	21
Informieren	21
Planen	21
Entscheiden	26
Realisieren	27
Kontrollieren	41
Auswerten (Fazit)	44
Benutzerhandbuch	45
Quellenverzeichnis	64
Glossar	65
Anhang	66

# **Teil 1**

## **IPA-Umfeld und -Ablauf**

# Teil 1: IPA-Umfeld und -Ablauf

## Aufgabenstellung

Die Gründe für dieses Projekt liegen in einem CMS (Content Management System), welches wir (Asperger AG) am Entwickeln sind. Das CMS wird vor allem mit PHP erstellt und soll bei Kundenanfragen verwendet werden. Ziel ist, dass andere gängige CMS, die wir in unserer Firma verwenden, abgelöst werden.

Der Formulargenerator soll unseren Kunden ermöglichen, selber ein Formular zu erstellen. Dafür wird eine Benutzeroberfläche benötigt, auf welcher man einfach Formulare verwalten kann. Auf der Benutzeroberfläche kann ein Formular erstellt, gelöscht, umbenannt, angepasst und kopiert werden. Eine Liste mit bestehenden Formularen ist auf der Seite ersichtlich.

Im Formularmodus kann aus unterschiedlichen Bausteinen ein eigenes Formular erstellt werden. Dazu gehören textarea, input, checkbox, radiobox und fileupload. Ausserdem sollen folgende Möglichkeiten für die Validierung des Formulars zur Auswahl vorhanden sein und dann auch beim Abschicken des Formulars geprüft werden.

### INPUT

**number** – Minimum und Maximum Wert

**text** – Email  
– Anzahl Zeichen  
– PLZ(Maximale Anzahl Zeichen definierbar)

**date** – Datum von  
– Datum bis

**textarea** – Anzahl Zeichen

**fileupload** – Maximale Dateigrösse in MB

**Für alle Felder** – Pflichtfelder(Müssen ausgefüllt werden)

Zudem gibt es Einstellungen für das jeweilige Formular, darauf können Empfänger, Betreff und die Antwortadresse definiert werden. Sollte ein Formular ausgefüllt abgeschickt werden, soll der eingetragene Empfänger eine E-Mail mit allen Angaben des Formulars erhalten. Dazu zählen auch die Dateien welche mit dem

Formular abgeschickt werden. Das Design der Benutzeroberfläche des Formulargenerators soll von Mike Bühler selbstständig gestaltet werden.

Als Resultat wird eine Ansicht der bestehenden Formularen erwartet, wobei man dort diese erstellen, löschen, ändern, anpassen und kopieren kann, sowie funktionierende Formulare, welche man ausfüllen kann und eine Email mit eingetragenen Daten an den gewählten Empfänger gesendet werden kann.

- |   |   |
|---|---|
| <b>Folgende Test sind durchzuführen</b> | <ul style="list-style-type: none"><li>- Senden des Emails bei abschicken eines Formulars</li><li>- Erstellen eines neues Formulars</li><li>- Anpassen eines bestehendes Formulars</li><li>- Löschen eines Formulars</li><li>- Kopieren eines Formulars</li><li>- Umbenennen eines Formulars</li></ul> |
|---|---|

Mike Bühler erstellt ein Benutzerhandbuch, welches den Formulargenerator erklärt und beschreibt. Eine zeitgemäße und zielgruppengerechte Gestaltung liegen ebenfalls in seinem Aufgabenbereich. Er legt dabei Wert auf eine verständliche Formulierung, da die Zielgruppe keinen technischen Hintergrund hat.

## FACHLICHE SCHWERPUNKTE

Die Umsetzung des Formulargenerators ist funktionsfähig und praxistauglich. Erfüllung der Anforderung für den Formulargenerator. Der Code ist lesbar geschrieben und ordentlich formatiert.

Die Umsetzung des Benutzerhandbuchs ist zielgruppen- respektive benutzergerecht. Das Benutzerhandbuch ist für Benutzer ohne technischen Hintergrund verständlich.

Gestalten der Benutzeroberfläche und des Benutzerhandbuchs für den Formulargenerator.

- |               |   |
|---------------|---|
| <b>Quelle</b> | <a href="https://host1.pkorg.ch/modules.php?name=IVIPA&amp;a=A10010">https://host1.pkorg.ch/modules.php?name=IVIPA&amp;a=A10010</a><br>(www.pkorg.ch) |
|---------------|---|

## Projektorganisation

Das Ziel der Arbeit ist ein funktionierender Formulargenerator zu erstellen.

Dazu brauche ich folgendes

- Arbeitswerkzeug: Laptop, Block und Schreibzeug für Skizzen, ruhiger Ort zum Arbeiten
- Mustererstellung: Photoshop
- Programmierung: Dreamweaver
- Gestaltung Formular: Dreamweaver
- Benutzerhandbuch: Word, InDesign
- Dokumentation, Zeitplanung: Word

Für das Projekt habe ich 96 bis 116 Stunden zur Verfügung.

Für die Erstellung des Projekts und die Dokumentation wende ich IPERKA an.

## Vorkenntnisse

Ich habe viel Erfahrung mit der Herstellung von Webseiten gemacht, schwerpunktmässig mit HTML und CSS. Ich erstelle Formulare mit HTML, CSS und Javascript. Mit Javascript kann ich die Grundlagen in diesen Bereichen anwenden: Zeichenketten, Verzweigungen, Schleifen, Umrechnung, Masse usw. Bildbearbeitung und Mustererstellung mache ich mit Photoshop. Mit Adobe InDesign erstelle ich Flyer, Visitenkarten und Broschüren. Für Videobearbeitung verwende ich das Programm Premiere Pro.

Ich verfüge über php Grundkenntnisse.

In meiner Lehrzeit erstellte ich auch Logos und Animationen. Für die Kundschaft der Asperger AG mache ich vorwiegend Instandhaltung und Erweiterungen/Ergänzungen ihrer Webseiten.

## Vorarbeiten

Folgende Vorarbeiten für die IPA wurden ausgeführt

- Entwurf Zeiterfassung
- Entwurf Titelblatt
- Layout für Dokumentation und Arbeitsjournal

## Firmenstandards

Für diese IPA gelten folgende Firmenstandards:

PHP Richtlinien siehe Seite 66 im Anhang

Namenskonventionen für Dateien siehe Seite 67 im Anhang

## Zeitplan (nächste Seite)

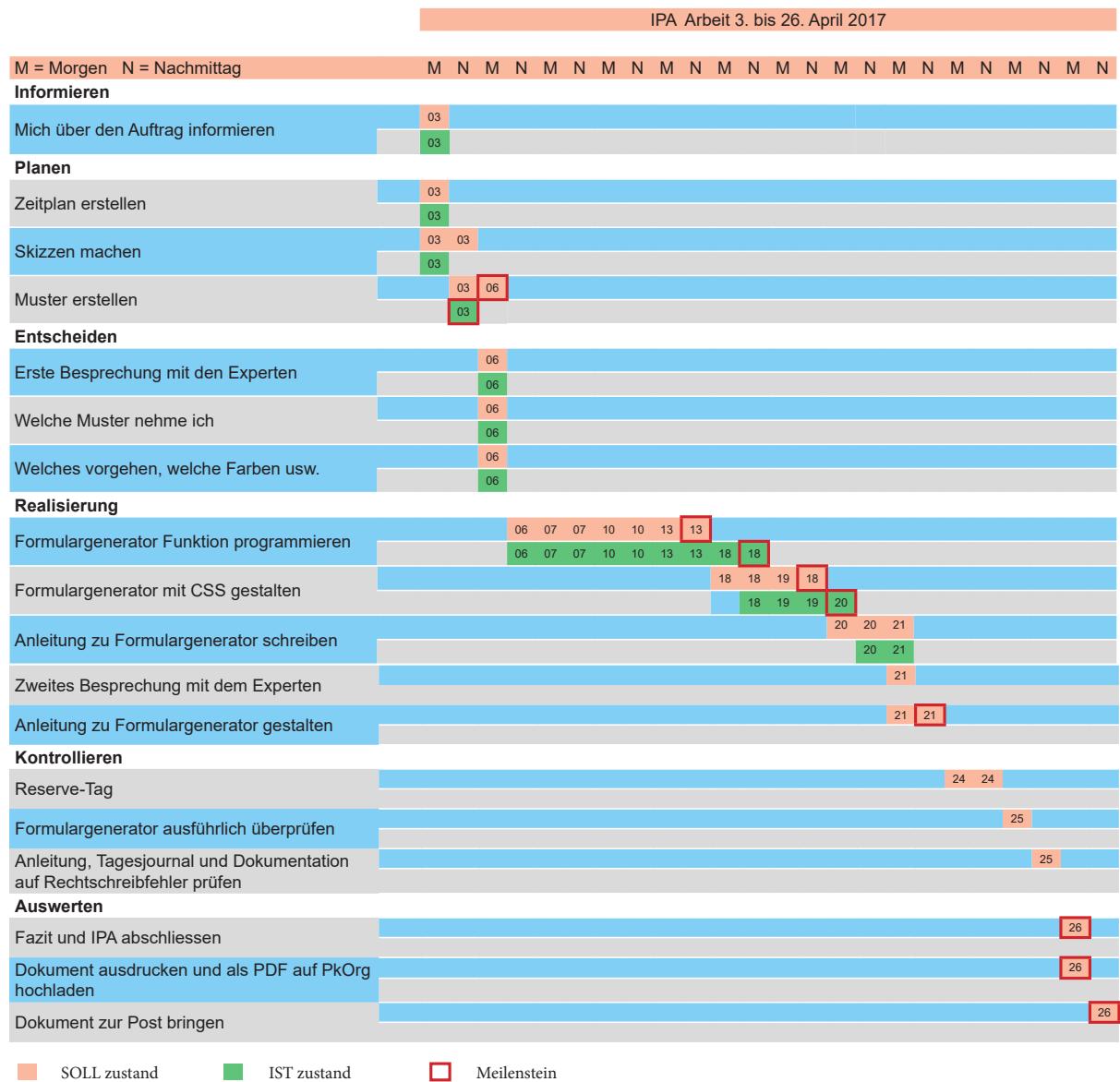
**Programm** InDesign CC

**Schriftart** Arial

**Schriftfarbe** schwarz 000000

**Farben**  
Lachs f8bea7  
Blau 82cff5  
Grau dadada  
Rot cd1619  
Grün 60c37a

## Zeitplan SOLL/IST Vergleich



**Tab. 1: Zeitplanung**

## Arbeitsprotokoll

Datum	03.04.2017
Ziel	Informieren über den Auftrag, Zeitplan erstellen Skizze erstellen und mit der Mustererstellung beginnen.
Am Morgen	<ul style="list-style-type: none"> <li>- Auftrag und PDF's nochmals durchgelesen</li> <li>- Zeitplan erstellt</li> <li>- Skizzen für Seiten erstellen</li> <li>- Photoshop für Minus- und Plussymbol</li> <li>- Muster angefangen</li> </ul>
Nachmittag	<ul style="list-style-type: none"> <li>- Muster erstellt</li> <li>- Dokumentation geschrieben</li> </ul>
Schwierigkeit	Der Formulargenerator hat viele Seiten. Von welchen mache ich eine Musterseite?
Lösungen	Ich entschied mich für drei Seiten, bei denen ich ein Muster erstelle. 1. Muster / Anfangsseite, 2. Muster / Persönliche Angabe vor dem Erstellen, 3. Muster ein erstelltes Formular
Zielerreichung	<ul style="list-style-type: none"> <li>- Ich habe mich über den Auftrag informiert.</li> <li>- Der Zeitplan ist erstellt.</li> <li>- Skizzen gemacht.</li> <li>- Muster fertig gestellt, bin leicht vor dem Zeitplan</li> </ul>
Fazit	Mir ist bewusst, dass heute der einfache Teil war, in den nächsten Tagen folgt der schwierigere Teil.
Soll/IST	Bin Vor dem Zeitplan
Zeitaufwand	8 h 30 min

Datum	06.04.2017
Ziel	ungefähr 25 % von der Funktion vom Formulargenerator erstellen
Am Morgen	<ul style="list-style-type: none"> <li>- Nachschauen was heute meine Tagesziele sind (ungefähr 25 % von der Funktion vom Formulargenerator erstellen und den Expertenbesuch überstehen/überleben.</li> <li>- Erstellen des Index.php</li> <li>- Vorbereitung für den Expertenbesuch</li> <li>- Erstellung des form.php</li> </ul>
Schwierigkeit	<ol style="list-style-type: none"> <li>1. Ich wusste nicht, wie ich diese Seite benennen soll</li> <li>2. Zeigte mir einen Fehler auf Linie 47 an bei der form.php</li> </ol>
Lösung	<ol style="list-style-type: none"> <li>1. Habe den Namen form.php gegeben.</li> <li>2. Ich habe vergessen eine Klammer zu schliessen, um den nervigen Fehler zu beheben habe ich 10 min verschwendet</li> </ol>
Expertenbesuch	Expertenbesuch: Beim Expertenbesuch ging es darum, uns gegenseitig kennen zu lernen. Herr Amsler Jean-Dominique fragte mich nach meinem Auftrag. Ebenfalls interessierte er sich für meinen Arbeitsplatz. Wir schauten gemeinsam den Zeitplan an. Er fragte mich nach einem geeigneten Termin für die Präsentation und das Fachgespräch. Mein Vorschlag war ab. 8. Mai 2017. Problem: war so nervös, dass ich nicht klar denken konnte, heisst auch, dass ich Mühe hatte mich auszudrücken.

Datum	06.04.2017
Nachmittag	<ul style="list-style-type: none"> <li>- funktion.php erstellt. Der Code ist vom cms kopiert</li> <li>- Erstellung des inputs</li> <li>- An der Dokumentation und Tagesjournal gearbeitet</li> </ul>
Zielerreichung	Würde sagen: ja aber knapp. Wäre gerne weitergekommen. Hatte Mühe mit dem logischen Denken beim Programmieren
Fazit	Ich machte mir unbegründete und zu viele Gedanken wegen dem Expertenbesuch
Soll/IST	Bin im Zeitplan
Zeitaufwand	8 h 30 min

Datum	07.04.2017
Ziel	Ungefähr 50% der Formulargenerator Funktion erstellen.
Am Morgen	<ul style="list-style-type: none"> <li>- Dokumentation: Ein Teil vom 06.04.2017 nachschreiben</li> <li>- Einbauen des Buttons: Funktion löschen</li> <li>- Dokumentation dazu schreiben</li> <li>- Das 2. Muster angepasst</li> <li>- Zeitplan am Ende etwas abgeändert und Meilensteine hinzugefügt (24.04.2017 als Reserve eingeplant/abgeändert).</li> <li>- Funktionen hinzugefügt. Das Geschriebene bleibt gespeichert, sobald der Anwender auf „weiter“ klickt. Bei einer Abänderung/Ergänzung muss nicht nochmals alles geschrieben werden.</li> </ul>
Schwierigkeit	Beim Code: <code>,onClick="confirm(,&lt;?php echo \$filename[,filename']; ?&gt; löschen?)"&gt;</code> erscheint das Feld nicht mit der Abfrage, ob es wirklich gelöscht werden soll
Lösung	<code>onClick="return confirm(,&lt;?php echo \$filename[,filename']; ?&gt; löschen?)"&gt;</code>
Nachmittag	Dokumentation schreiben
Zielerreichung	Ja bei Formulargenerator Funktion habe ich das Ziel erreicht. Man kann ein leeres Formular erstellen und auch wieder löschen. Die Benutzer-Daten werden gespeichert und der Benutzer kann es im Nachhinein nochmals bearbeiten. Die Dokumentation habe ich nicht ganz fertig geschrieben. Diese werde ich am 08.04.2017 nachschreiben
Fazit	Dies war ein guter Tag. Ich kam zügig vorwärts.
Soll/IST	Mit dem Formulargenerator bin ich im Zeitplan. Dokumentation muss nachgeführt werden
Zeitaufwand	7 h 15 min

Datum	08.04.2017
Ziel	Nacharbeit Dokumentation schreiben und Korrekturen.
Nachmittag	Arbeit an der Dokumentation
Zielerreichung	Ja, die Dokumentation ist nachgeschrieben und die Rechtschreibung wurde geprüft und korrigiert
Hilfe durch Drittperson	Hilfestellung bei der Rechtschreibung durch Käthy Bühler
Zeitaufwand	2 h 00 min

Datum	10.04.2017
Ziel	Ungefähr 75 % – 85 % der Funktion des Formulargenerators fertig stellen.
Am Morgen	<ul style="list-style-type: none"> <li>– Arbeitsjournal und Dokumentation nochmals durchgelesen und auf Fehler prüfen</li> <li>– Mich über das Tagesziel orientieren</li> <li>– Arbeitsjournal eintragen</li> <li>– Grundaufbau der Seite form.php</li> <li>– Erstellung der text.php</li> <li>– Erstellung der Seite textarea</li> <li>– Erstellung der Seite email.php</li> <li>– Erstellung der Seite date.php</li> <li>– Erstellung der Seite fileuploads.php</li> <li>– Erstellung der Seite number.php</li> <li>– Erstellung der Seite select.php (ohne die Funktion Option)</li> </ul>
Nachmittag	<p>Erstellung der Seite radiobox.php      Erstellung der Seite checkbox.php      Funktionen bei formview.php programmieren      Arbeitsjournal und Dokumentation schreiben</p>
Zielerreichung	Bei der Programmierung habe ich das Ziel fast erreicht. Ich hätte gerne noch das Problem mit der „Option“ gelöst. Ich wollte mich damit nicht zu lange aufhalten und verschob die Problemlösung auf den Donnerstag. Mit der Dokumentation bin ich nicht ganz fertig geworden. Nach meiner Schätzung bin ich leicht hinter dem Zeitplan. Ich kann den Zeitverlust vielleicht am Donnerstag aufholen.
Fazit	An diesem Tag kam ich mit der Arbeit zügig voran, bis an das Problem mit der Option.
Soll/IST	Bin knapp im Zeitplan, eher leicht im Verzug
Zeitaufwand	9 h 00 min

Datum	13.04.2017
Ziel	Mein heutiges Ziel ist die Funktion abschliessen zu können
Am Morgen	<ul style="list-style-type: none"> <li>– Arbeit an der Dokumentation. Projektorganisation, Wirtschaftlichkeitsanalyse und Zielpublikumsanalyse schreiben.</li> <li>– Ich erstelle für select/checkbox und radiobox.php eine weitere Seite (option/check und radio.php). Somit ist das Problem vom Vortag behoben.</li> <li>– Ich verbinde text.php, textarea.php, email.php, date.php, fileuploads.php, number.php und select.php mit dem input.</li> </ul>
Schwierigkeit	Die Aufgabenstellung soll den Auftrag beschreiben, aber es gehört auch in den Bereich Information, weil ich das IPERKA verwende. Entweder füge ich es doppelt ein oder ich muss mich auf eins davon entscheiden.
Lösungen	Die detaillierte Aufgabenstellung lasse ich so stehen. Unter dem Titel Information schreibe ich eine Zusammenfassung der Aufgabenstellung
Nachmittag	<ul style="list-style-type: none"> <li>– Ich machte ein paar Funktionsvereinfachungen/ Namensabkürzungen. Zum Beispiel statt name='datefrom' und name='dateto' nur noch name='max' name='min'</li> <li>– Testen ob alles funktioniert</li> <li>– Versuche das Problem zu beheben</li> <li>– Tagesjournal und Dokumentation</li> </ul>

Datum	13.04.2017
Schwierigkeit	Bei der Checkbox ist ein Problem aufgetaucht. Es kann nur eine Checkbox hinzugefügt werden und wenn ich eine neue erstellen möchte, wird das Vorherige immer wieder überschrieben.
Lösungen	Noch keine
Zielerreichung	Das Tagesziel habe ich nicht erreicht. Ich habe ein Problem bei der Checkbox. Der Anwender kann momentan nur eine Checkbox erstellen. Wenn er eine weitere erstellen möchte, wird das Vorherige immer wieder überschrieben. Das Problem konnte ich noch nicht lösen. Dazu fehlen noch die Ausgabe und die E-mail-Versendung. Ich denke, dass ich am 18.04.2017 bis Mittag fertig werde mit der Funktion.
Fazit	Problem vom Vortag relativ schnell gelöst, dafür habe ich nun ein anderes Problem, dass mich schon viel Zeit gekostet hat.
SOLL/IST	Ich bin hinter dem Zeitplan. Funktion sollte fertig sein
Zeitaufwand	7 h 35 min

Datum	18.04.2017
Ziel	Laut meiner Planung sollte ich heute mit der Gestaltung des Formular-generators beginnen. Weil ich mit der Funktion noch nicht fertig bin, ist es nun mein Ziel, dies vor dem Mittag fertig zu stellen.
Am Morgen	<ul style="list-style-type: none"> <li>- Fehlersuche geht weiter</li> <li>- Radiobox und select getestet und den Fehler behoben (sozusagen der gleiche wie bei Checkbox)</li> <li>- Die Ausgabe erstellt</li> <li>- Ausgabe getestet</li> <li>- An der Dokumentation gearbeitet</li> </ul>
Lösung vom Vortag	Den Fehler habe ich gefunden: ein Tippfehler, das Zeichen ; vergessen einzufügen und ein Logikfehler.
Hilfe durch Drittperson	Ich musste mir Hilfe von Florian holen um den fehler zu beheben
Nachmittag	<ul style="list-style-type: none"> <li>- An der Funktion E-Mail versenden gearbeitet</li> <li>- Gestaltung mit CSS für den Formulargenerator</li> <li>- Arbeitsjournal und Dokumentation</li> </ul>
Zielerreichung	Das Ziel, den Fehler zu beheben habe ich erreicht. Leider habe ich doch noch bis 14:10 an der Funktion gearbeitet. Das zu lange war. Bin jetzt hinter dem Zeitplan.
Fazit	Nach einer ruhelosen Nacht war ich sehr froh, den Fehler gefunden zu haben. Fehler bei der Programmierung können schwerwiegende Zeitverluste nach sich ziehen und den Zeitplan durcheinander bringen.
SOLL/IST	Bin hinter dem Zeitplan
Zeitaufwand	8 h 30 min

Datum	19.04.2017
Ziel	Formulargestaltung fertig stellen
Am Morgen	<ul style="list-style-type: none"> <li>- Gestaltung des Formulargenerators</li> <li>- Bei allen Seiten Code (div) und Label hinzugefügt</li> </ul>
Nachmittag	<ul style="list-style-type: none"> <li>- Suche nach einem geeigneten E-Mailversand</li> <li>- Den Code anpassen</li> <li>- Dokumentation und Arbeitsjournal</li> </ul>
Zielerreichung	Ich konnte heute die Gestaltung abschliessen und einen kleinen Teil der Anleitung schreiben. Bin noch hinter dem Zeitplan, weil ich vorhatte, mindestens die Hälfte der Anleitung zu schreiben. Ich habe leider nur ungefähr 20 % geschafft
Fazit	Ich bin jetzt bis auf das Testen des Formulargenerators fertig. Es gibt noch kleinere Verbesserungen zu machen, aber die mache ich noch am Schluss, wenn ich Zeit habe. Ich bin fast wieder im Zeitplan.
SOLL/IST	Bin hinter dem Zeitplan. Gestaltung Formular sollte Fertig sein
Zeitaufwand	8 h 30 min

Datum	20.04.2017
Ziel	Formulargestaltung beenden und min. 20 % der Anleitung zu schreiben
Am Morgen	<ul style="list-style-type: none"> <li>- Die Formulargestaltung fertig gestellt</li> </ul>
Nachmittag	<ul style="list-style-type: none"> <li>- Gestaltung verfeinert (z.B. Umrandung hinzugefügt) davon sind 30 min Mittag</li> <li>- Alle Minus- und Plus-Symbole durch Text ersetzt (passten nicht mehr so gut zum Formulargenerator)</li> <li>- Geeignetes Symbol für den Button „ZURÜCK“ für die Startseite gesucht und eingebaut.</li> <li>- Begonnen die Anleitung zu schreiben</li> <li>- An der Dokumentation und am Arbeitsjournals gearbeitet. Ein Rechtschreibfehler beim Programmierungs-Text korrigiert. (von Pflichtfeld auf Pflichtfeld).</li> </ul>
Zielerreichung	Ich konnte heute die Gestaltung abschliessen und einen kleinen Teil der Anleitung schreiben. Bin noch hinter dem Zeitplan, weil ich vorhatte, mindestens die Hälfte der Anleitung zu schreiben. Ich habe leider nur ungefähr 20 % geschafft
Fazit	Ich brauchte für die Gestaltung des Formulargenerators mehr Zeit, als ich gedacht habe. Ich verlor viel Zeit bei den diversen Verbesserungen.
SOLL/IST	Bin hinter dem Zeitplan.
Zeitaufwand	8 h 30 min

Datum	21.04.2017
Ziel	Anleitung fertig schreiben und gestalten
Am Morgen	<ul style="list-style-type: none"> <li>- Anleitung weitergeschrieben</li> <li>- Besuch des Experten</li> <li>- Anleitung fertig geschrieben</li> </ul>
Schwierigkeit	Beim Vorführen des Formulargenerators hatte der Formulargenerator einige kleinere Fehler die ich vorher nicht gehabt hatte oder nicht aufgefallen sind.
Nachmittag	<ul style="list-style-type: none"> <li>- Mehre kleinere Fehler beim Formulargenerator noch behoben.</li> </ul>
Zielerreichung	Das Ziel habe ich heute nicht erreicht, sollte mit der Gestaltung der Anleitung fertig sein. Ich habe mit diesem nicht mal begonnen. Ziel nicht erreicht.
Fazit	Es war nicht so ein guter Tag für mich. Der Formulargenerator hatte bei der Vorführung kleinere Probleme. Das führte dazu, dass ich diese noch heute beseitigen musste. Zum Glück habe ich einen Reservetag eingeplant
SOLL/IST	Bin hinter dem Zeitplan. Anleitung Sollte Fertig sein
Zeitaufwand	6 h 50 min

Datum	22.04.2017
Ziel	Dokumentation Korrekturen.
Am Morgen	Dokumentatration auf Rechtschreibfehler Prüfen
Hilfe durch Drittperson	Hilfestellung bei der Rechtschreibung durch Käthy Bühler
Zielerreichung	Die Dokumentation wurde korrigiert.
Zeitaufwand	1 h 30 min

Datum	24.04.2017
Ziel	Den Reservetag nutzen, um wieder in den Zeitplan zu kommen oder sogar vor dem Zeitplan zu kommen
Am Morgen	<ul style="list-style-type: none"> <li>- Gestaltung der Anleitung gemacht</li> <li>- Formular Generator getestet und dokumentiert</li> </ul>
Nachmittag	<ul style="list-style-type: none"> <li>- Am Arbeitsjournal und der Dokumentation gearbeitet</li> </ul>
Zielerreichung	Habe mein Ziel erreicht, bin wieder im Zeitplan
Fazit	Bin erleichtert, dass ich im Zeitplan liege
SOLL/IST	Bin dank des Reservetages im Zeitpan
Zeitaufwand	8 h 30 min

Datum	25.04.2017
Ziel	Dokumentation abschliessen
Am Morgen	– Arbeiten an der Dokumentation
Nachmittag	– Arbeiten an der Dokumentation
Zielerreichung	Dokumentation ist abgeschlossen, soweit möglich
Fazit	Die Dokumentation ist extrem zeitaufwendig
SOLL/IST	Bin im Zeitplan
Zeitaufwand	8 h 30 min

Datum	26.04.2017
Ziel	Dokumentation kontrollieren und gegebenenfalls korrigieren
Am Morgen	– Alles nochmals kontrolliert, ausgedruckt und hochgeladen
Nachmittag	– Dokumentation binden lassen und zur Post bringen
Zielerreichung	Die Arbeit wurde hochgeladen und die Dokumentation wurde verschickt
Fazit	Bin echt froh, dass ich es jetzt abgeschlossen habe
SOLL/IST	Bin im Zeitplan
Zeitaufwand	8 h 30 min

## Organisation der Arbeitsergebnisse

Die Entwicklung und Dokumentation mache ich lokal auf meinem Laptop. Ich arbeite mit drei verschiedenen Speicherorten. Speicherort Nummer 1 ist lokal auf meinen Laptop. Zusätzlich mache ich fast alle zwei Stunden eine Sicherung auf meiner externen Festplatte. Bevor ich am Abend nach Hause gehe, gibt es noch eine Absicherung auf dem FTP Server.

## Wirtschaftlichkeitsanalyse

Ja, der Formulargenerator ist wirtschaftstauglich. Ein Kunde oder eine Privatperson kann ein Formular erstellen und verwenden. Dank einer kleinen, benutzerfreundlichen Anleitung ist es für den Anwender nicht schwer, ein eigenes Formular anzufertigen. Das fertige Formular ist in schwarz/weiss und kann per Email versendet werden. Auch dies trägt zur Tauglichkeit des Produktes bei. Dazu kann man das Formular individuell anpassen und spart somit Zeit.

Dank dem Formulargenerator können wir in Zukunft einfacher und schneller Formulare erstellen und an die Kunden versenden. Auch sind die Formulare immer im gleichen Format.

## Zielpublikumsanalyse

Die Zielgruppe sind hauptsächlich Kunden der Asperger AG die selbstständig ein Formular mit Hilfe einer Anleitung erstellen wollen. Zum Beispiel, wenn eine Firma einen Kurs ausschreibt mit einem Anmeldeformular. Dieses Formular kann mit dem Formulargenerator generiert werden. Der Anbieter des Kurses setzt im Formular die entsprechenden Bausteine ein, zum Beispiel für Name, Adresse, Email, Datum, Ort, usw. Das ausgefüllte Formular wird per Email versendet.

## Teil 2

# Projektdokumentation

# Teil 2: Projekt-Dokumentation

## Management Summary

### AUSGANGSLAGE

Mein IPA-Projekt besteht darin, einen Formulargenerator und das dazugehörige Benutzerhandbuch zu erstellen. Der Grund für die Wahl dazu war, dass die Asperger AG ihr gängiges CMS (Content Management System) ablösen will. Ich habe bis jetzt noch nie einen Formulargenerator programmiert und gestaltet. Darum war ich zu Beginn der Arbeit zugleich neugierig, aber auch nervös. Die Aufgabenstellung verwirrte mich am Anfang, nahm aber schnell klare Formen an. Ich Motiviert und überzeugt, dass ich diese Aufgabe korrekt erstellen kann, ging ich an das Werk.

### VORGEHEN

Für die Realisierung wendete ich IPERKA an. Für die Programmierung benutzte ich die Applikation DreamWeaver. Das Benutzerhandbuch schrieb ich mit Word, fügte es am Schluss in die Applikation InDesign ein. Nachdem mir klar war, was meine Aufgabe ist, fertigte ich einen Zeitplan an. Am Anfang kalkulierte ich die Zeit knapp, damit ich am Schluss noch einen Zeitpuffer habe.

Ich entschied mich relativ schnell, wie der Aufbau der Seiten für den Formulargenerator aussehen sollte: übersichtlich, einfach und bedienungsfreundlich.

Bei der Realisierung des Formulargenerators begann ich mit der Seite index.php. Dann folgten die Seiten form.php; formview.php; input.php

<b>Index.php</b>	Startseite mit einem Button (Formular erstellen)
<b>Form.php</b>	Grunddatenseite mit zwei Buttons, zwei Pflichtfelder (Formularname; Email) Nach der Programmierung der form.php, zurück auf die Indexseite und die Buttons; Formular erstellen; Anschauen; Bearbeiten; Einstellungen; Kopieren und Löschen einbauen.
<b>Formview.php</b>	zwei Buttons einbauen (Auswahl; Zurück)
<b>Input.php</b>	die verschiedenen Buttons programmieren und einbauen: Textfeld; Text; Email; Radiobox; Checkbox; Datum; Nummer; Auswahlliste; Datei hochladen und Zurück.

Je nach Funktionalität der Buttons programmierte ich einzelne Felder wie Pflichtfeld; max./min -; von/bis - Eingabefelder. Ebenfalls der Funktionbutton: Weiter Die Seiten radiobox.php, checkbox.php und select.php sind komplexer aufgebaut. Hier benötigt es je eine weitere Seite: radio.php; check.php und option.php.

**radiobox php** Programmierung der Eingabefelder und Buttons. Dabei werden auf dieser Seite zusätzliche Buttons aktualisiert, sobald man auf den Button „speichern“ klickt. (z.B. Radiobox erstellen) Dies gilt auch für die Programmierung der Seiten check.php und option.php.

**Hinweis** Die Codes für die einzelnen Funktionen sind im Anhang unter Glossar auf Seite 65 zu finden. Die Darstellung der Buttons und Eingabefelder, sowie die detailliertere Vorgehensweise befinden sich von Seite 27 bis 39.

Das Benutzerhandbuch gestaltete ich mit wenigen Worten, dafür mit vielen Bildern, die die einzelnen Seiten zeigen. Siehe dazu das Benutzerhandbuch auf Seite 45.

## ERGEBNIS

Mit dem **Formulargenerator** kann man nun ein Formular erstellen, indem man auf die Startseite <http://localhost/IPA/formgenerator/index.php> geht. Von dieser Seite aus gelangt man automatisch auf die nächste Seite, mit den Eingabefeldern: Formularname und Email. Das sind Pflichtfelder, ohne Ausfüllen dieser Felder ist es nicht möglich weiter zu fahren und ein Formular zu erstellen. Die folgenden Seiten sind selbsterklärend aufgebaut. Der Anwender hat die Wahl, welche Elemente er benötigt. Er bestimmt, ob ein Eingabefeld pflichtig oder eine PLZ eingefügt werden soll. Er kann bestimmen wie gross die Anzahl Zeichen sein darf und Elemente für Datum, Nummern/Zahlen, Email, Maximum – Minimum und von – bis verwenden. Bei jedem Element, das er verwendet, gibt er eine Bezeichnung ein. Diese kann er abspeichern. Mit dem Löschbutton kann er erstellte Elemente oder auch erstellte Formulare löschen. Mit dem Button Bearbeiten werden Korrekturen und Ergänzungen vorgenommen. Mit dem Button Datei hochladen besteht die Möglichkeit, Dateien hochzuladen und zu verschicken. Der Button Anschauen hat die Funktion, dass der Anwender das zu erstellende Formular ansehen und bei Fertigstellung verschicken kann. Der Anwender kann ebenfalls einzelne Elemente kopieren und eine Auswahlliste erstellen.

Das **Benutzerhandbuch** besteht aus einem Titelblatt, Inhaltsverzeichnis, Einleitung und die Anleitung für die Benutzung des Formulargenerators.

#### Fazit

Der Formulargenerator ist funktionstüchtig und kundenfreundlich aufgebaut, das Benutzerhandbuch strukturiert und anschaulich gestaltet. Die Probleme bei der Programmierung haben mich viel Zeit gekostet, was anschliessend zu den Abweichungen führte. Mir ist klar, dass bei der Gestaltung des Formulargenerators mehr möglich gewesen wäre. Gerne hätte ich für jede Seite einen ersichtlichen Namen gegeben, damit der Anwender immer weiss, wo er sich befindet. Auch beim Benutzerhandbuch bemerkte ich zu spät, dass es sinnvoll gewesen wäre eine Musterseite eines erstellten Formulars einzufügen. Grössere Probleme bereiteten mir die Gestaltung und das Schreiben der Dokumentation. Ich habe Mühe, mich sowohl schriftlich wie auch mündlich gut auszudrücken. Darum schrieb, korrigierte und änderte ich immer wieder. Das hat viel Zeit gekostet. Nun ist es vollbracht, bin zwar ausgelaugt aber glücklich zugleich, die Arbeit abzuschliessen.

#### Zielsetzung und Abweichungen

Ziel	Abweichung	Grund
Muster erstellen bis am 06.04.17	Erstellt am 03.04.17	Ich kam zügig vorwärts
Funktionalität programmieren bis am 13.04.17	Fertig programmiert am 18.04.17	Probleme bei der Programmierung der Seiten radiobox.php; checkbox.php und select.php. Die Gründe sind; Zeichen; vergessen; Tippfehler und Logikfehler
Gestaltung des Formulargenerators bis am 19.04.17	Fertig gestaltet am 20.04.17	Wegen der Suche nach den Programmierfehlern kam ich in Zeitverzug
Fertigstellung der Anleitung bis am 21.04.17	Fertig gestellt am 24.04.17	Ebenfalls wegen dem erwähnten Fehler
Einzelne Seiten beschriften, damit der Anwender sieht, wo er sich befindet	Keine Beschriftung	Keine Zeit
Beim Benutzerhandbuch eine Musterseite einbauen	Keine Musterseite eines erstellten Formulars	Keine Zeit, zu spät realisiert, dass ich eine Musterseite einbauen sollte, damit der Anwender ein Endprodukt sieht.

**Tab. 2: Zielsetzung und Abweichung**

# Formulargenerator

## INFORMIEREN

Funktionalität	Ich erstelle einen Formulargenerator, der es dem Kunden ermöglicht, selber ein Formular zu erstellen. Dafür programme ich eine webbasierte Benutzeroberfläche, auf der die Formulare verwaltet werden können. Auf der Webseite kann man ein Formular erstellen, löschen, umbenennen, anpassen und kopieren. Bereits bestehende Formulare müssen ersichtlich sein. Schlussendlich muss gewährleistet sein, dass das erstellte Formular samt eingetragenen Daten per Email verschickt werden kann.
Gestaltung	Die Gestaltung der Benutzeroberfläche soll übersichtlich, kundenfreundlich sein und einen logischen Aufbau haben.
Benutzerhandbuch	Zum Formulargenerator schreibe ich eine einfache Anleitung mit Bildern, die es dem Anwender erleichtert, ein Formular zu erstellen und/oder zu ändern.

## PLANEN

Zeitplanung	Jedes Projekt benötigt eine sorgfältige Zeitplanung, damit die Frist für die Fertigstellung eingehalten werden kann. Ebenfalls müssen Meilensteine und Ist/Soll-Zeitaufwand ersichtlich sein. (siehe dazu Zeitplanung im ersten Teil der Dokumentation).  Für die Zeitplanung habe ich am Anfang die Zeit knapp berechnet. Meine Überlegung war, falls unerwartete Probleme auftauchen und ich dadurch hinter den Zeitplan falle, ich noch genügend Zeit-Reserve habe. So kann ich ohne grossen Zeitdruck das Versäumte nachholen und bin beruhigter, wenn ich weiss, dass gegen Ende noch genügend Zeit vorhanden ist. Wahrscheinlich habe ich die Erstellung für die Funktion des Formulars zu knapp berechnet. Ich vermute, dass ich dort hinter den Zeitplan fallen könnte. Die Dokumentation sowie das Tagesjournal habe ich extra nicht in die Zeitplanung genommen. Das Journal schreibe ich etappenweise, zum Beispiel bei unerwarteten Problemen, Verzögerungen oder positiven Erfolgsergebnissen und Abschluss und Neuanfang weiterer Aufgaben. Die Dokumentation schreibe ich immer gegen Ende des IPA Tages oder/und Zuhause.
-------------	---

## Skizzen erstellen

Skizzen und Entwürfe sind notwendig, damit man die ersten Gedanken und Vorstellungen auf Papier bringt und abschätzt, ob sie brauchbar und umsetzbar sind. Bei den Skizzen habe ich mich nur auf das Wesentliche beschränkt und wollte nicht allzu viel Zeit dafür aufwenden. Deshalb ist nur eine Skizze vorhanden. Sie beschreibt den Aufbau des Formulargenerators.

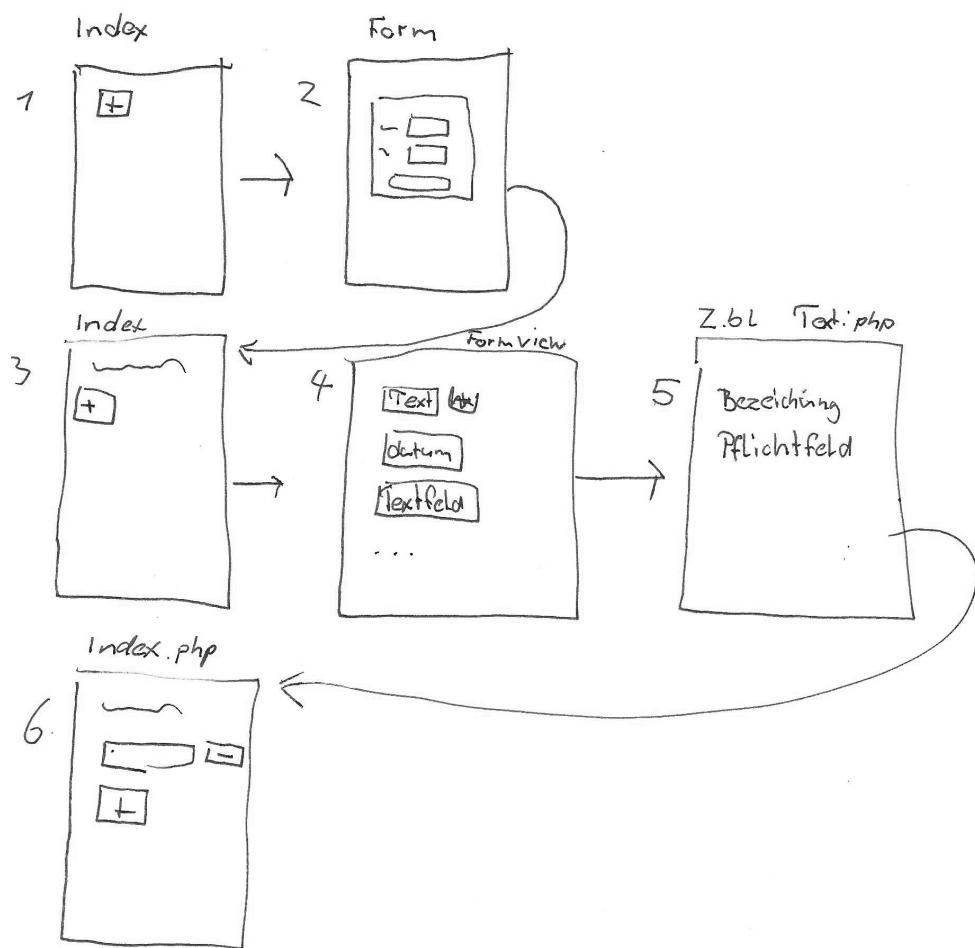


Abb. 1: Skizze. So stelle ich mir den Aufbau des Formulargenerators vor

## Muster erstellen

Ich erhalte mit einem Muster einen Einblick, wie die Seite auf einem Anwender wirken kann. Dabei werden Farbe, Grösse und Darstellung dargestellt.

Gleich zu Beginn bei der Musterherstellung hatte ich ein Problem. Ich überlegte mir, zu welcher Seite ich ein Muster erstellen sollte. Grund, der Formulargenerator hat viele Seiten. Ich entschied mich für folgende Seiten:

**Muster 1:**  
Startseite  
(Index.php)

Obwohl die Seite fast leer ist, ist sie mit einer Startseite einer Webseite zu vergleichen.

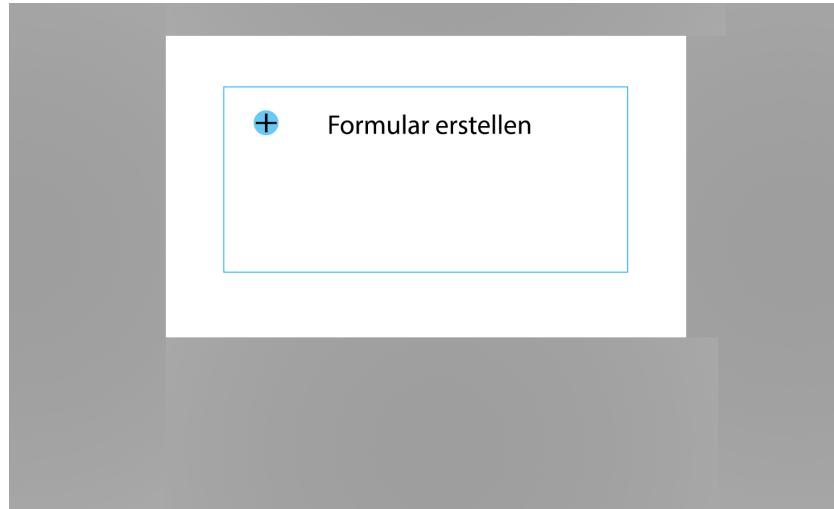


Abb. 2: Variante 1A

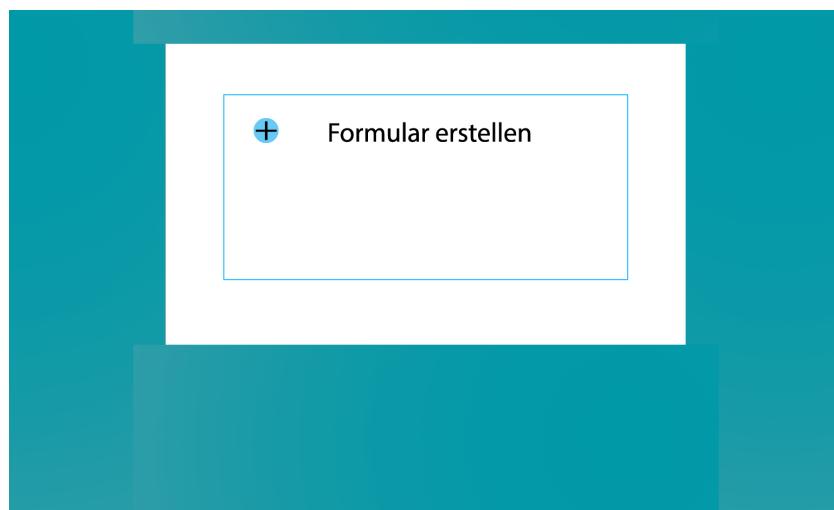


Abb. 3: Variante 1B

**Muster 2: Grund-datenerfassung  
(Form.php)**

Das zweite Muster zeigt die Seite, in welcher der Ersteller/-in seine persönlichen Daten eingeben muss, bevor er ein Formular generieren kann.

The screenshot shows a form interface with a light gray background. In the center, there is a white rectangular box with a thin blue border containing five input fields. Each field has a label above it: "Formular Name:", "E-Mail:", "Antwortadresse:", "Betreff:", and "Max Dateigrösse:". Below these fields are two buttons: a blue "Weiter" button and a blue "Zurück" button. The entire form is centered on a dark gray background.

Abb. 4: Variante 2A

The screenshot shows a form interface with a dark teal background. In the center, there is a white rectangular box with a thin blue border containing five input fields. Each field has a label above it: "Formular Name:", "E-Mail:", "Antwortadresse:", "Betreff:", and "Max Dateigrösse:". Below these fields are two buttons: a yellow "Weiter" button and a yellow "Zurück" button. The entire form is centered on a dark teal background.

Abb. 5: Variante 2B

**Muster 3:**  
**Beispiel-**  
**formular**

Das dritte Muster zeigt, wie ein generiertes Formular am Ende aussehen könnte.

Formular

Anrede:  Frau  Herr

Name: Name Vorname

E-Mail: Beispiel@bluewin.ch

Tele 0683412671

Anzahl

Bemerkung

**Abb. 6: Variante 3A**

Die Musterseiten habe ich mit dem Programm Photoshop kreiert.

Formular

Anrede:  Frau  Herr

Name: Name Vorname

E-Mail: Beispiel@bluewin.ch

Tele 0683412671

Anzahl

Bemerkung

**Abb. 7: Variante 3B**

Weiss:	FFFFFF
Schwarz:	000000
Hellgrau:	f2f2f2
Grau:	a0a0a0
Blau:	67c9f6
Türkis:	0099a8
Gelb:	d1b400

## ENTSCHEIDEN

<b>Entscheid Muster</b>	Ich wähle bei den Mustern 1 und 2 die Variante B mit den Farben Türkis und Gelb. Bei Muster 3 die Variante A.
<b>Grund für Variante B</b>	Muster 1A und 2A sind zu fade mit grau und blau. Besonders der graue Hintergrund macht die Seite langweilig. Bei Muster 1B und 2B sind die Farben reizvoller mit Türkis und Gelb. Zusätzlich sind es auch die Farben, die bei der Asperger AG verwendet werden. Eigentlich war dies der Grund für die Wahl der Farben.
<b>Grund für Variante A</b>	Bei Muster 3 ziehe ich die Variante schwarz-weiss vor. Es kommt einem alltäglichen Formular näher und ist neutraler.
<b>Schriftart</b>	Myriad Pro. Diese Schrift ist gut lesbar und erscheint mir passend.
<b>Mein Vorgehen</b>	Ich werde den Formulargenerator im Dreamweaver programmieren, weil ich dieses Programm sehr gut kenne. Dazu maskiert es den Code und gibt mir so zusätzliche Hilfestellungen.  Ich bau zuerst die Funktionalität des Formulargenerators auf. Dort fange ich mit der Index Seite an. Der nächste Schritt ist, dass ein Formular generiert und ebenfalls das erstellte Formular wieder gelöscht werden kann. Wenn diese Funktionen abgeschlossen sind, folgt die Funktionalität mit den Auswahlfeldern, wo der Nutzer die Möglichkeit hat, das Formular selber zusammen zu stellen. Anschliessend beginne ich mit der Gestaltung des Formulars. Wenn das Formular gestalterisch und funktional abgeschlossen ist, wird die dazugehörige Benutzeranleitung angefertigt. Diese werde ich zuerst in Word schreiben. Für die Rechtschreibungüberprüfung eignet sich Word gut. Wenn die Anleitung geschrieben ist, werde ich sie in InDesign gestalten und den Text von Word in InDesign kopieren. Der letzte Schritt ist dann alles nochmals zu prüfen.

## REALISIEREN

**Allgemeines** Für die Programmierung des Formulargenerators verwende ich die Applikation Dreamweaver CC. Ich werde eine gewisse Reihenfolge der Programmierung einhalten, damit ich laufend die Funktionalität prüfen kann. Die Programmiercodes für die einzelnen Funktionen befinden sich im Anhang.

**Englisch** Ich vergab bei der Programmierung englische Namen, weil:

- Englische Namen sind meist kürzer als die deutschen Bezeichnungen
- Ich kann ö, ä und ü vermeiden
- Englisch verstehen die meisten Personen

Der Code sollte selbsterklärend sein

The screenshot shows a web form with a teal border. At the top, there is a radio button group for 'Anrede' (Address) with 'Frau' (Female) and 'Herr' (Male), where 'Herr' is selected. Below it is a text input field for 'Name' containing 'Muster'. The next row has a text input field for 'Vorname' containing 'Hans'. The third row contains an email input field with the value 'h.muster@ebmnet.ch'. A section titled 'Kurse' (Courses) follows, with two checkboxes: 'Englisch' (English) and 'Japanisch' (Japanese), where 'Japanisch' is checked. Below this is a date input field with the value '01.05.2017'. A dropdown menu labeled 'select' shows 'Anfänger' (Beginner) as the selected option, with 'Erfahren' (Experienced) as the other choice. At the bottom, there is a file upload input field with the placeholder 'Bitte Foto hinzufügen' (Please add photo) and a 'Browse...' button, showing the message 'No file selected.'. A large yellow button at the bottom right is labeled 'Abschicken' (Send).

Abb. 8: Endprodukt

Auf dieser Seite ist das erstellte Formular ersichtlich. Mit dem eingebauten Button ABSCHICKEN kann man nun die Email versenden.

## Indexseite erstellen

**index.php:** Diese Seite ist die Startseite der Homepage:

<http://localhost/IPA/formgenerator/index.php>

Hier startet der Anwender mit der Erstellung eines Formulars. Die Programmierung ging erstaunlicherweise sehr gut.



**Abb. 9: Haussymbol: wird auf jeder Seite erscheinen.**

**Es bringt den Anwender zurück auf die Startseite**

Damit ich auf der Indexseite weiter fahren kann, muss ich zuerst die Seite mit den Grunddaten erstellen, damit ich das Programmierte testen und anwenden kann.

## Grunddatenseite erstellen

**form.php:** Auf dieser Seite muss der Anwender zuerst seine Emailadresse und den Namen des Formulars eingeben. Man könnte diese Seite auch mit Name/Vorname, Adresse ergänzen, was aber nicht zu meinem Auftrag gehört. Ich habe placeholder="..." angewendet um den Benutzer darauf hinzuweisen, was er in die Felder reinschreiben soll.

A screenshot of a web page with a light blue header bar. In the top right corner of the header is a small house icon. Below the header is a form area with five input fields. Each field has a label above it and a placeholder inside. The fields are: "Formularname\*" (placeholder: "Name des Formulars"), "E-Mail\*" (placeholder: "E-Mail"), "Antwortadresse\*" (placeholder: "Name, Adresse"), "Betreff\*" (placeholder: "Betreff"), and "Max. Dateigrösse in MB" (placeholder: "Dateigrösse"). Below the form are two large yellow buttons with the text "Weiter" and "Zurück". The rest of the page is white with no other content.

**Abb. 10: Formular erstellen: diese Seite hat Pflichtfelder (required) Ebenfalls programmierte ich die Buttons WEITER und ZURÜCK**

- Button WEITER** Mit Klick auf WEITER wird man automatisch weitergeleitet auf die Indexseite. Der eingegebene Formularname bleibt gespeichert.
- Button ZURÜCK** Mit Klick auf ZURÜCK wird das Eingegebene nicht gespeichert.
- Buttons vorbereiten** **index.php:** Programmierung der Buttons FORUMLAR ERSTELLEN, BEARBEITEN und LÖSCHEN. Mit Klick auf den Button WEITER auf der Grunddatenseite gelangt man nun auf die nächste Seite. Hier erscheinen nun der Button FORUMLAR ERSTELLEN und der Formularname, den ich auf der Grunddatenseite eingegeben habe. Test erfolgreich.

Die Buttons ANSCHAUEN, KOPIEREN und BEARBEITEN kann ich noch nicht testen, weil diese Seiten noch nicht vorhanden sind.

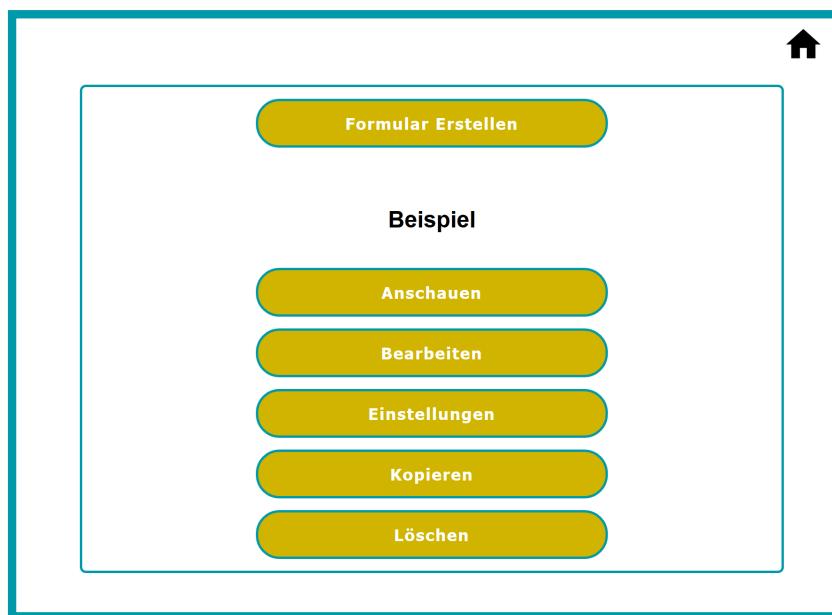


Abb. 11: Formularübersicht: diese Seite hat Pflichtfelder (required) Ebenfalls programmierte ich die Buttons WEITER und ZURÜCK.

<b>Erklärung zu Button</b>	<p><b>Anschauen:</b> Hier kann der Anwender das erstellte Formular anschauen und per Email versenden</p> <p><b>Bearbeiten:</b> Einzelne Bausteine erstellen, zusammenfügen, eingegebene Texte korrigieren</p> <p><b>Einstellungen:</b> Grunddaten bearbeiten (Emailadresse und Name des Formulars)</p> <p><b>Kopieren:</b> Wird für das Duplizieren der Formulare benutzt</p> <p><b>Löschen:</b> Mit einem Klick kann man erstellte Formulare löschen. Damit man nichts versehentlich löscht, baute ich eine Sicherheitsabfrage ein, die nachfragt, ob wirklich gelöscht werden soll. (return confirm)</p>
<b>Automatische Abspeicherung</b>	Damit die eingegebenen Daten des Anwenders gespeichert bleiben, wird es in den Ordner Form abgespeichert
<b>Seite für erstellte Bausteine</b>	formview.php programmieren. Hier erscheinen die erstellten Bausteine

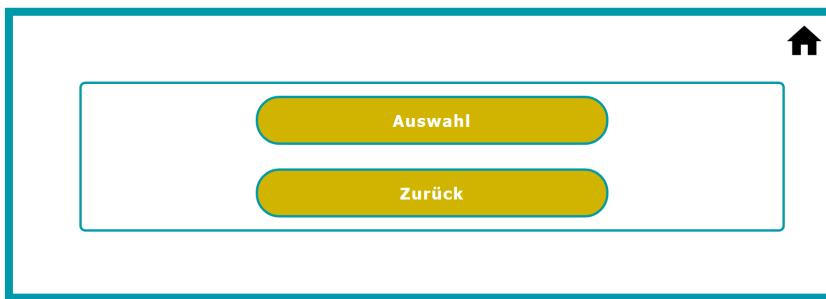


Abb. 12: Mit Klick auf AUSWAHL kommt man auf die Inputseite

Elementeauswahl [input.php](#) programmieren.

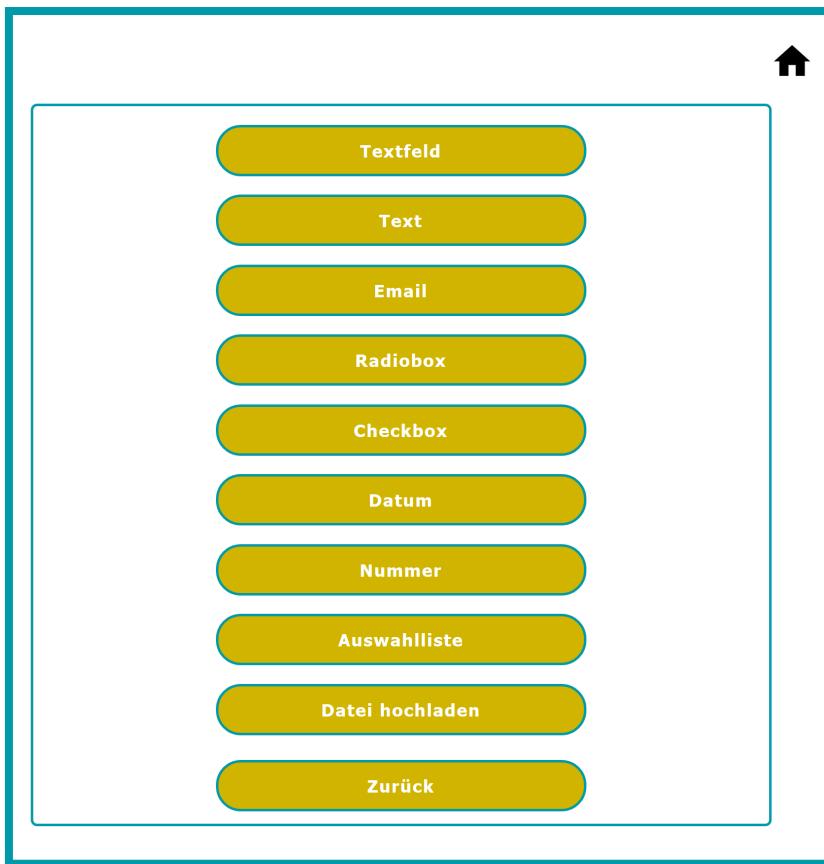


Abb. 13: Auswahl aller Inputs

Button Textfeld

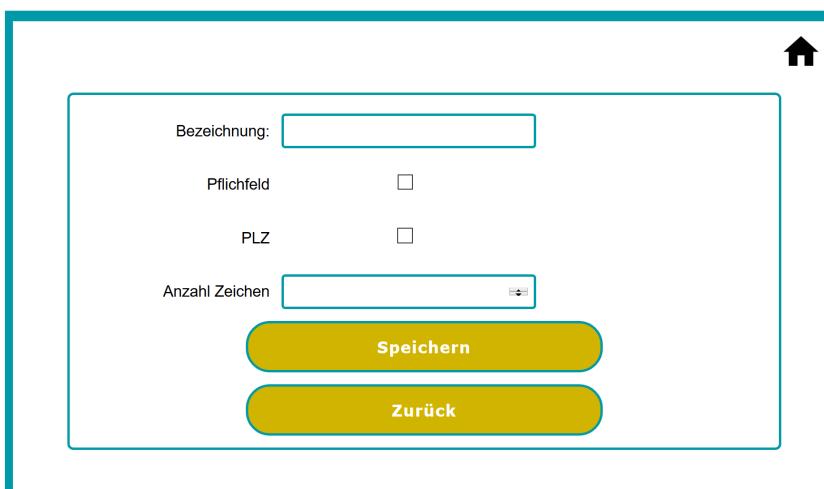


Abb. 14: Eingabe des Textfeldes

**Button Text**

The screenshot shows a form interface with a teal border. At the top right is a black house icon. Inside the form, there is a text input field labeled "Bezeichnung:" with a placeholder. Below it is a checkbox labeled "Pflichtfeld". A horizontal input field labeled "Max Zeichen" is followed by a character count indicator. At the bottom are two yellow rounded rectangular buttons labeled "Speichern" and "Zurück".

**Abb. 15: Eingabe des Textes**

**Button Email**

The screenshot shows a form interface with a teal border. At the top right is a black house icon. Inside the form, there is a text input field labeled "Bezeichnung:" with a placeholder. Below it is a checkbox labeled "Pflichtfeld". At the bottom are two yellow rounded rectangular buttons labeled "Speichern" and "Zurück".

**Abb. 16: Eingabe des Feldes Email**

**Button  
Radiobox**

**radiobox.php** programmieren: Feld mit Bezeichnung Buttons SPEICHERN und ZURÜCK. Ebenfalls baue ich einen versteckten Button RADIOBOX ERSTELLEN ein. Dieser erscheint, sobald man auf SPEICHERN klickt.

The screenshot shows a form interface with a teal border. At the top right is a black house icon. Inside the form, there is a text input field labeled "Bezeichnung:" with the placeholder "Anrede". Below it is a yellow rounded rectangular button labeled "Speichern". Underneath is another yellow rounded rectangular button labeled "Zurück". At the bottom is a third yellow rounded rectangular button labeled "Radiobox Erstellen".

**Abb. 17: Eingabe der Radiobox**

Weil mit der Radiobox mehrere Auswahlelemente erstellt werden können, ist es notwendig, dass eine weitere Seite programmiert werden muss.

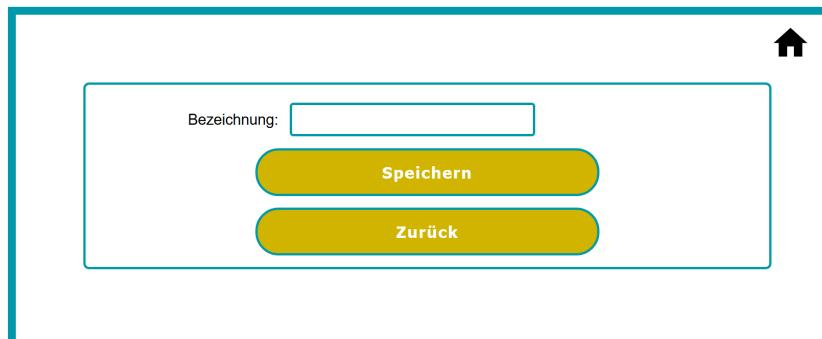


Abb. 18. Bezeichnung der Radiobox

#### Button Radiobox erstellen

radio.php programmieren: das Feld Bezeichnung, sowie die Button SPEICHERN und ZURÜCK. Mit Klick auf SPEICHERN kommt man zurück auf die Seite Radiobox. Das erstellte Radioelement ist sichtbar. Ebenfalls habe ich den Button LÖSCHEN eingefügt, damit man die einzelnen Elemente löschen kann.



Abb. 19: Hinzufügen der Radiobox

**Button Checkbox** [checkbox.php](#) programmieren: Feld mit Bezeichnung Buttons SPEICHERN und ZURÜCK. Ebenfalls bau e ich einen versteckten Button CHECKBOX FELDER ERSTELLEN ein. Dieser erscheint, sobald man auf SPEICHERN klickt.



Abb. 20: Eingabe der Checkbox

Weil mit der Checkbox mehrere Auswahlelemente erstellt werden können, ist es notwendig, dass eine weitere Seite programmiert werden muss.

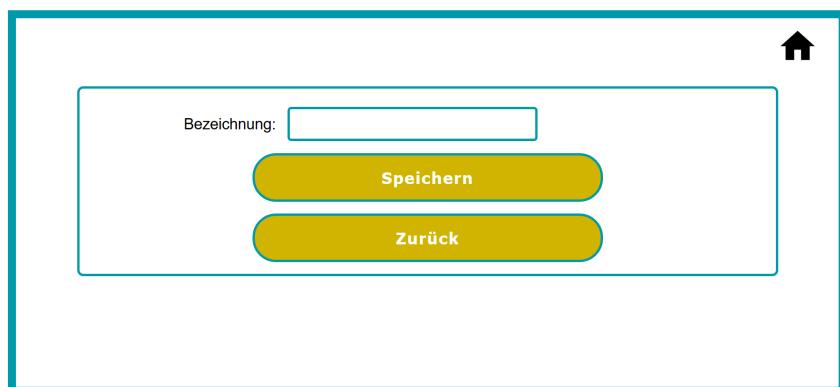


Abb. 21: Bezeichnung der Checkbox

**Button Checkbox  
erstellen**

[check.php](#) programmieren: das Feld Bezeichnung, sowie die Button SPEICHERN und ZURÜCK. Mit Klick auf SPEICHERN kommt man zurück auf die Seite Checkbox. Das erstellte Checkelement ist sichtbar. Ebenfalls habe ich den Button LÖSCHEN eingefügt, damit man die einzelnen Elemente löschen kann.

The screenshot shows a web page with a light blue header bar containing a house icon. Below the header is a white rectangular form area. At the top left of the form is a label 'Bezeichnung:' followed by a text input field containing the placeholder 'Auswahl'. To the right of the input field is a yellow button labeled 'Speichern'. Below it is another yellow button labeled 'Zurück'. Further down is a yellow button labeled 'Checkbox Felder Erstellen'. This is followed by a yellow button labeled 'Einzelzimmer Checkbox'. At the bottom of the form is a yellow button labeled 'Löschen'.

Abb. 22: Hinzufügen der Checkbox

Der Unterschied zwischen Radiobox und Checkbox ist folgender: Bei Radiobox kann man von verschiedenen Möglichkeiten nur eines auswählen, indem man auf das zugehörige Kreissymbol klickt. Bei Checkbox kann man hingegen mehrere Möglichkeiten anklicken.

**Button Datum**

Felder mit Bezeichnung; Pflichtfeld; Datum von; Datum bis; Buttons SPEICHERN und ZURÜCK

The screenshot shows a web page with a light blue header bar containing a house icon. Below the header is a white rectangular form area. At the top left is a label 'Bezeichnung:' followed by a text input field. To the right of the input field is a small square checkbox labeled 'Pflichtfeld'. Below the input field is a yellow button with a checkmark icon. Next is a label 'Datum von:' followed by a text input field containing the placeholder 'dd.mm.jjjj'. Below it is another label 'Datum bis:' followed by a text input field containing the placeholder 'dd.mm.jjjj'. To the right of the 'Datum von:' field is a small square checkbox. At the bottom of the form are two yellow buttons labeled 'Speichern' and 'Zurück'.

Abb. 23: Eingabe des Feldes Datum

**Button Nummer** Felder mit Bezeichnung; Pflichtfeld; min. Wert; max. Wert; Buttons SPEICHERN und ZURÜCK

The screenshot shows a user interface for entering a number. At the top right is a house icon. Inside the main area, there is a label 'Bezeichnung:' followed by a text input field containing 'Nummer'. Below it is a checkbox labeled 'Pflichtfeld'. Underneath are two input fields for 'Min. wert' and 'Max. wert', each with a small info icon. At the bottom are two buttons: a yellow rounded rectangle labeled 'Speichern' and a blue rounded rectangle labeled 'Zurück'.

Abb. 24: Eingabe des Feldes Nummer

**Button Auswahlliste**

`select.php` programmieren: Feld mit Bezeichnung

Buttons SPEICHERN und ZURÜCK. Ebenfalls bau e ich einen versteckten Button AUSWAHL ERSTELLEN ein. Dieser erscheint, sobald man auf SPEICHERN klickt.

The screenshot shows a user interface for selecting a list. At the top right is a house icon. Inside the main area, there is a label 'Bezeichnung:' followed by a text input field containing 'Kursort'. Below it are three yellow rounded rectangle buttons labeled 'Speichern', 'Zurück', and 'Auswahl Erstellen' from top to bottom.

Abb. 25: Eingabe der Auswahlliste

Weil mit der Auswahlliste mehrere Auswahlelemente erstellt werden können, ist es notwendig, dass eine weitere Seite programmiert werden muss.

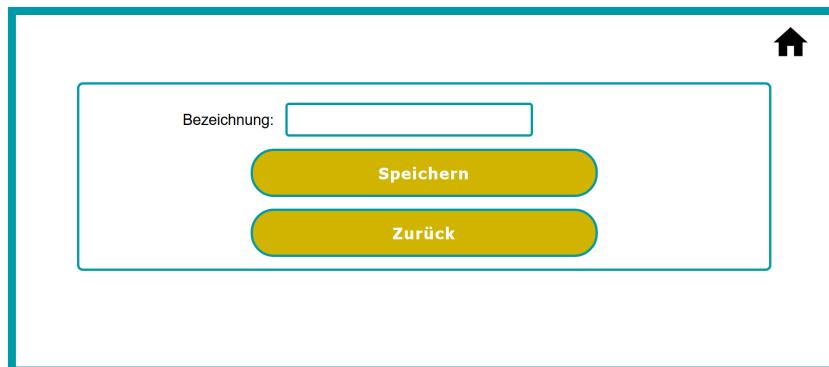


Abb. 26: Bezeichnung der Auswahlliste

#### Button Auswahl erstellen

[option.php](#) programmieren: das Feld Bezeichnung, sowie die Button SPEICHERN und ZURÜCK. Mit Klick auf SPEICHERN kommt man zurück auf die Seite Auswahlliste. Das erstellte Auswahlelement ist sichtbar. Ebenfalls habe ich den Button LÖSCHEN eingefügt, damit man die einzelnen Elemente löschen kann.

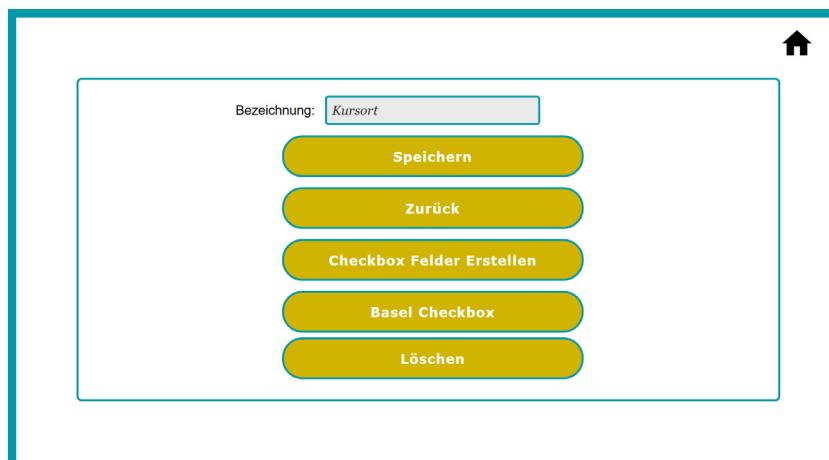


Abb. 27: Hinzufügen eines weiteren Elementes

**Button Datei  
Hochladen**

Felder mit Bezeichnung; Pflichtfeld; Buttons SPEICHERN und ZURÜCK

The screenshot shows a user interface element for file uploads. It consists of a large rectangular input field with a teal border. Inside the field, there is a placeholder text 'Bezeichnung:' followed by a small rectangular input field with a teal border. Below this, the word 'Pflichtfeld' is followed by a small square checkbox. At the bottom of the input field, there are two rounded rectangular buttons with a yellow gradient background and black text. The top button is labeled 'Speichern' and the bottom one is labeled 'Zurück'. In the top right corner of the entire interface, there is a small house-shaped icon.

Abb. 28: Eingabe des Feldes Datei hochladen

## FORMULARGESTALTUNG

Allgemein

Ich mache die Gestaltung mit der CSS.

Ich verwende zwei verschiedene CSS Dateien; eines für das erstellte Formular in der Ansicht; das andere für die restlichen Seiten.

Die Programmierung zur Gestaltung ist im Anhang ersichtlich

Gestaltung  
Ansicht

Das CSS für die Ansicht mache ich bis auf den Button ABSCHICKEN schwarz-weiss, weil die meisten Formulare in schwarz-weiss gehalten werden. Ich versende bei der Programmierung „div“, damit die Felder, die zum hineinschreiben benutzt werden, in der vertikalen Mitte bleiben. „Form“ benutzte ich um dem Formular eine gewisse Grösse zu geben. Bei der Schrift gab ich als Standard Grösse 13 px und die Schriftart ist Arial, „Times New Roman“, Times, serif;. Bei den Eingabefeldern verwendete ich die Schriftart Georgia, „Times New Roman“, Times, serif; um zu erkennen was ausgefüllt wurde.

Gestaltung  
Seiten

Ich gestaltete die Buttons so, wie ich diese im Muster angefertigt habe. Ich liess aber die Plus- und Minuszeichen weg und ersetze diese mit Text (z.B. alle Minuszeichen mit der Bezeichnung „Löschen“ ersetzt.)

Die Buttons haben eine türkisblaue Umrundung, der Hintergrund ist gelb, die Schrift weiss fett. Sobald man mit dem Mauszeiger über den Button fährt, wird der Hintergrund weiss, die Schrift wird schwarz fett und die Umrundung bleibt Türkis.



Abb. 29: Ansichtsbeispiel der Buttons, aktiv und inaktiv

Um dem Anwender klar zu verdeutlichen, bei welchem Button er gerade ist, wird die Schrift in schwarz fett dargestellt. Die gesamte Grösse des Formulars macht ich 820 px mit wrapper. Alles was darüber hinaus ist, wird in Türkisfarbe angezeigt. Der Grund dazu ist, dass es dem Anwender erleichtert wird, das Dokument zu erstellen und zu lesen. Wenn der Anwender ein Eingabefeld ausgewählt hat, wird dieses in hellgrau dargestellt. So weiss der Nutzer stets welches Feld er ausgewählt hat oder eben nicht.



Abb. 30: Ansichtsbeispiel der Textfelder, aktiv und inaktiv

Verwendete Farben	Schwarz:	0000005
	Weiss:	FFFFFFF
	Hellgrau	eaeaea
	Buttonfarbe	
	Türkis	#0099a8
	Gelb	#d1b400
	Schwarz:	000000
	Weiss:	FFFFFFF
Schrift	Arial	
	Georgia	
CSS Name	Formularstyle.css	
	style.css	

## BENUTZERANLEITUNG

Allgemein	Die Benutzeranleitung gestaltete ich mit Adobe InDesign. Die fertiggestellte Anleitung befindet sich im Anhang. Ich verwendete viele Bilder, damit der Anwender weiss, auf welcher Seite er eine Eingabe machen muss.
Titelseite	Die Gestaltung ist eine Eigenkreation. Verwendete Farben Türkis #0099a8 Gelb #d1b400 Schwarz: 000000 Weiss: FFFFFFFF
Struktur	Ich wählte 13 Schritte für die Anleitung. Dabei verwendete ich immer Bilder dazu.
Verwendete Farben	Anleitung: Türkis #0099a8 Gelb #d1b400 Schwarz 000000 Weiss FFFFFFFF
Schrift	Franklin Gothic Book / Medium

## Kontrollieren

### KONTROLLE FORMULARGENERATOR

	Ja	Nein
Leitet jeder Link zu den richtigen Seiten	x	
Werden die Nutzerdaten übernommen	x	
Müssen die Pflichtfelder ausgefüllt werden	x	
Wird es angezeigt, wenn ein Pflichtfeld ausgefüllt werden muss	x	
Wird das Ausgewählte korrekt auf das Formular übertragen	x	
Kann das Formular wieder gelöscht werden	x	
Können die einzelnen Elemente gelöscht werden	x	
Funktioniert die Checkbox	x	
Funktioniert die Radiobox	x	
Funktioniert Select (Auswahlliste)	x	
Funktioniert das Nummernfeld	x	
Funktioniert das Textfeld	x	
Funktioniert das Feld für Text	x	
Funktioniert der Button DATEI HOCHLAGEN	x	
Funktioniert die max. Grösse für Datei hochladen	x	
Funktionieren Datumfelder (Bezeichnung, max. und min)	x	
Funktioniert das E-Mail Feld	x	
Funktioniert der Button ZURÜCK korrekt	x	
Werden die Pflichtfelder bei der Auswahl übernommen	x	
Funktioniert das Kopieren	x	
Funktioniert der Button EINSTELLUNG korrekt und bleiben die eingegebenen Daten gespeichert	x	
Kommt man mit dem Symbol – Haus – auf jeder Seite wieder zurück auf die Startseite	x	
Kann das Formular mit dem Button ANSCHAUEN betrachtet werden	x	
Funktioniert das Versenden der E-Mails	x	

**Tab. 3:**  
**Kontrolle Formulargenerator**

### AUFRAGSTEST

	Ja	Nein
Senden des Emails bei Abschicken eines Formulars	x	
Erstellen eines neuen Formulars	x	
Anpassen eines bestehenden Formulars	x	
Löschen eines Formulars	x	
Kopieren eines Formulars	x	
Umbenennen eines Formulars	x	

**Tab. 4:**  
**Auftragstest**

## KONTROLLE ANLEITUNG

	Ja	Nein
Inhaltsverzeichnis vorhanden	x	
Formulargeneratoranleitung hat die richtige Reihenfolge	x	
Bilder sind passend zum Text und helfen zum Verständnis	x	
Formulierung kontrolliert	x	
Prüfen auf Rechtschreibfehler	x	
Ist alles in der gleichen Schrift geschrieben	x	
Ist alles vorhanden	x	

**Tab. 5:**  
**Kontrolle**  
**Anleitung**

## KONTROLLE DOKUMENTATION

	Ja	Nein
Ist es nach IPERKA gestaltet	x	
Ist Teil 1 und Teil 2 getrennt	x	
Deckblatt vorhanden	x	
Titelblatt vorhanden	x	
Aufgabenstellung vorhanden	x	
Projektorganisation vorhanden	x	
Vorkenntnisse dokumentiert	x	
Vorarbeiten erwähnt	x	
Firmenstandarts aufgezählt	x	
Zeitplan mit ist und soll vorhanden	x	
Arbeitsjournal vollständig	x	
Fremde Hilfe ist erwähnt	x	
Organisation der Arbeitsergebnisse ist dokumentiert	x	
Wirtschaftsanalyse ist dokumentiert	x	
Zielpublikumanalyse ist vorhanden	x	
Summary	x	
Fazit geschrieben	x	
Quellenverzeichnis gemacht	x	
Glossar	x	
Formulierungen kontrolliert	x	
Auf Rechtschreibfehler geprüft	x	

**Tab. 6:**  
**Kontrolle**  
**Dokumentation**

## QUALITÄTSSICHERUNG (durchgeführt am 24.04.2017)

Nr.	Aufgabe	Erwartetes Resultat	Eingetroffenes Resultat
1.1	Erstellen eines neuen Formular	Formular wird neu erstellt	Wie erwartet
1.2	Anpassen eines bestehendes Formular	Formular wird angepasst	Wie erwartet
1.3	Kopieren eines bestehendes Formular	Formular wird kopiert	Wie erwartet
1.4	Formular löschen	Formular wird gelöscht	Wie erwartet
2.1	Beim Formular Textfeld erstellen	Textfeld wird erstellt	Wie erwartet
2.2	Anpassen des Textfeldes	Textfeld wird angepasst	Wie erwartet
2.3	Beim Formular Text erstellen	Text wird erstellt	Wie erwartet
2.4	Anpassen des Textes	Text wird angepasst	Wie erwartet
2.5	Beim Formular Email erstellen	Email wird erstellt	Wie erwartet
2.6	Anpassen des Emails	Email wird angepasst	Wie erwartet
2.7	Beim Formular Datum erstellen	Datum wird erstellt	Wie erwartet
2.8	Anpassen des Datum	Datum wird angepasst	Wie erwartet
2.9	Beim Formular Datei hochladen erstellen	Datei hochladen wird erstellt	Wie erwartet
2.10	Anpassen Datei hochladen	Datei hochladen wird angepasst	Wie erwartet
3.1	Beim Formular Radiobox erstellen	Radiobox wird erstellt	Wie erwartet
3.2	Anpassen Radiobox	Bezeichnung Radiobox wird angepasst	Wie erwartet
3.3	Radiobox hinzufügen	Radiobox wird hinzugefügt	Wie erwartet
3.4	Hinzugefügte Radiobox anpassen	Radiobox wird angepasst	Wie erwartet
3.5	Beim Formular Checkbox erstellen	Checkbox wird erstellt	Wie erwartet
3.6	Anpassen Checkbox	Bezeichnung Checkbox wird angepasst	Wie erwartet
3.7	Checkbox hinzufügen	Checkbox wird hinzugefügt	Wie erwartet
3.8	Hinzugefügte Checkbox anpassen	Checkbox wird angepasst	Wie erwartet
3.9	Beim Formular Select erstellen	Select wird erstellt	Wie erwartet
3.10	Anpassen Select	Bezeichnung Select wird angepasst	Wie erwartet
3.11	Option hinzufügen	Option wird hinzugefügt	Wie erwartet
3.12	Hinzugefügte Option anpassen	Option wird angepasst	Wie erwartet
4.1	Bei Pflichtfeld nichts angeben	Formular wird nicht gesendet	Wie erwartet
4.2	Bei Datum kein Datum angeben	Formular wird nicht gesendet	Wie erwartet
4.3	Bei Datum kein Datum angeben	Formular wird nicht gesendet	Wie erwartet
4.4	Bei Email keine Email angeben, darf aber nicht leer sein	Formular wird nicht gesendet	Wie erwartet
4.5	Formular ausfüllen und versenden	Eine Email erhalten mit den ausgefüllten Daten	Wie erwartet
5.1	Bei Select das Option löschen	Option wird gelöscht	Wie erwartet
5.2	Alle Inputs löschen	Alle Inputs werden gelöscht	Wie erwartet

**Tab. 7: Qualitätssicherung**

## Auswerten (Fazit)

### WAS WÜRDE ICH DAS NÄCHSTE MAL ANDERS MACHEN?

**Anleitung:** Bei der Anleitung würde ich ein Musterformular erstellen, mit den verschiedenen Elementen und beschriften. Am Ende des Benutzerhandbuchs einfügen.

**Formulargenerator:** Jeder Seite einen Namen geben. Einstellung - Startseite – Grunddaten – Erstellte Elemente – Auswahl Elemente – Textfeld – Text – E-Mail – Radiobox – Auswahl Radiobox – Checkbox – Auswahl Checkbox – Datum – Nummer – Auswahlliste – Auswahl Auswahlliste – Datei hochladen.

**Dokumentation:** Ich würde die Dokumentation von Anfang weg strukturierter schreiben.

**Arbeitsjournal:** Ich machte das Arbeitsjournal mit detaillierter Zeitangabe, was überflüssig war.

### MIT WAS BIN ICH BESONDERS ZUFRIEDEN?

**Anleitung:** Die Anleitung habe ich mit vielen Bildern gestaltet.

**Formulargenerator:** Radiobox, Checkbox und Auswahlliste waren relativ aufwendig zum Programmieren. Ich bin sehr zufrieden, dass diese Elemente funktionieren.

**Dokumentation:** Mit der Dokumentation bin ich generell nicht ganz zufrieden, weil ich mich grundsätzlich mündlich sowie auch schriftlich nicht gut ausdrücken kann. Mit der Gestaltung bin ich hingegen zufrieden.

**Arbeitsjournal:** Das Journal habe ich konsequent jeden Tag detailliert geführt.

### WAS HABE ICH DARAUS GELERNT?

Das Schreiben der Dokumentation ist sehr zeitaufwendig. Ebenfalls muss man bei der Programmierung damit rechnen, dass sich ein Fehler einschleichen kann. Das lange Suchen führt ebenfalls zu Zeitverlust. Positiv ist, dass ich mehrere Codes php dazu gelernt habe.

Grundsätzlich machte mir die Erstellung des Formulargenerators Spass – nur das Schreiben der Dokumentation dazu war ein Muss.

# **Benutzerhandbuch**

## **Formulargenerator**

# Inhaltsverzeichnis

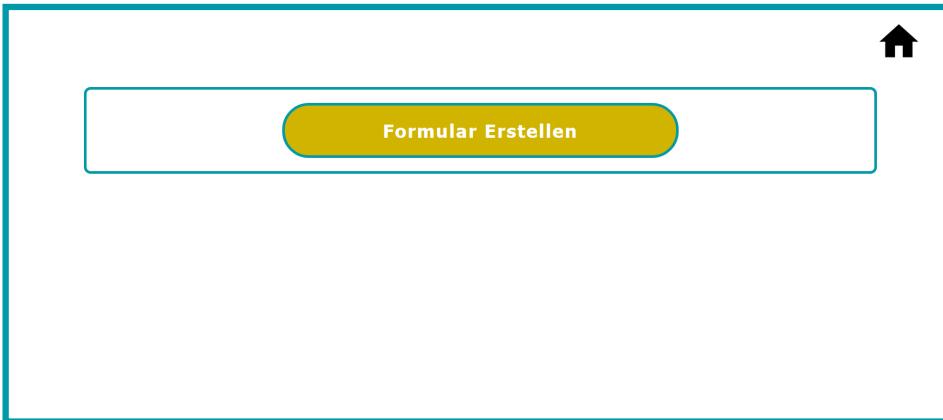
01. Grundeingaben	03
02. Formular bearbeiten	04
03. Auswahl	05
04. Texfeld erstellen	06
05. Text	07
06. Email	08
07. Radiobox	09
08. Checkbox	11
09. Datum	13
10. Nummer	14
11. Auswahlliste	15
12. Datei Hochladen	17
13. Einzelne Elemente löschen oder bearbeiten	18

## Einleitung

Mit diesem Programm können Sie selber Ihre Formulare erstellen, ergänzen und löschen. Dieses Handbuch vermittelt Ihnen in kurzen und einfachen Schritten die Vorgehensweise dazu. Loggen Sie sich dazu auf die Seite:  
<http://localhost/IPA/formgenerator/index.php> ein. Und los geht's.

## Anleitung Formular erstellen

Loggen Sie sich ein auf die Homepage Site: <http://localhost/IPA/formgenerator/index.php>  
Sie befinden sich nun auf der Startseite.



Wenn sie das haben, wählen sie den Button **FORMULAR ERSTELLEN**



## Grundeingaben

The screenshot shows a web-based form titled "Grundeingaben". It contains five input fields: "Formularname\*", "E-Mail\*", "Antwortadresse\*", "Betreff\*", and "Max. Dateigröße in MB". Below the fields are two buttons: "Weiter" (in yellow) and "Zurück" (in light blue). A small house icon is located in the top right corner of the form area.

- Beim Formularnamen können sie dem Formular einen beliebigen Namen geben.
- Bei unserem Muster lautet der Name \* Beispiel \*
- Geben Sie Ihre E-Mailadresse an
- Geben Sie die Antwortadresse an
- Geben Sie den Betref an
- Geben Sie den maximale Grösse der Datei an.
- Klicken Sie auf **WEITER**

Hinweis: Die Felder müssen zwingend ausgefüllt sein, damit Sie auf die nächste Seite kommen.  
Mit dem Button **ZURÜCK** bleiben die eingegebenen Daten gespeichert.

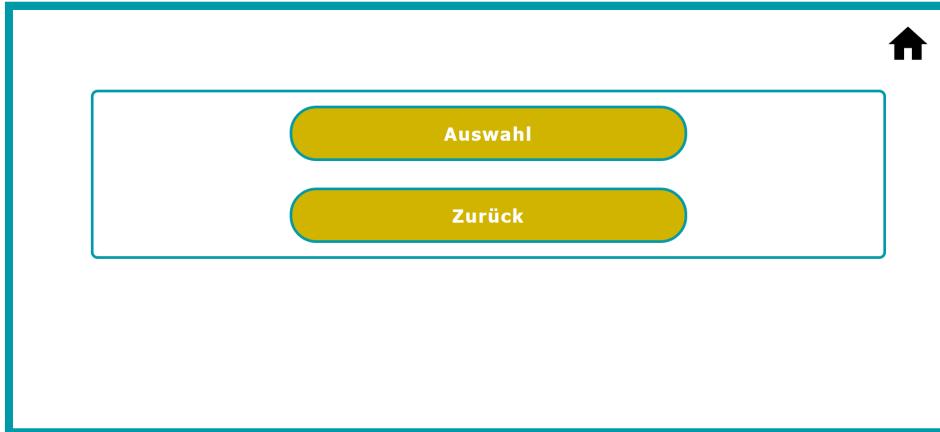
The screenshot shows a web interface titled "Formular Erstellen". At the top, there is a button labeled "Formular Erstellen". Below it, under the heading "Beispiel", are five buttons: "Anschauen", "Bearbeiten", "Einstellungen", "Kopieren", and "Löschen".

Folgende Funktionen stehen Ihnen zur Verfügung:

- Mit dem Button **ANSCHAUEN**, können sie ihr erstelltes Formular betrachten.
- Mit dem Button **BEARBEITEN** fügen Sie Elemente ein und können Korrekturen vornehmen
- Mit der **EINSTELLUNG** werden Änderungen und Korrekturen bei den Grundeingaben vorgenommen.
- Falls Sie das Formular duplizieren wollen, klicken Sie auf den Button **KOPIEREN**.
- Mit **LÖSCHEN** wird das Formular gelöscht.

## Formular bearbeiten

Auf **BEARBEITEN** klicken:



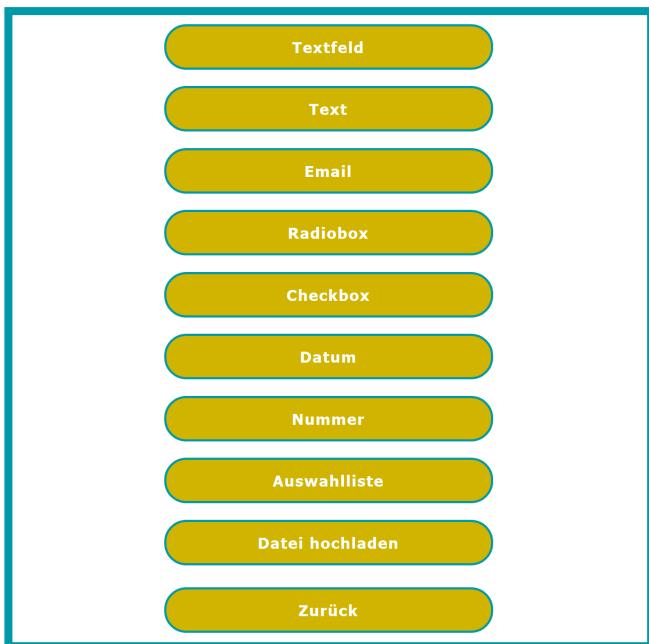
Mit dem **AUSWAHL**-Button können sie Elemente auswählen.

Klicken sie nun auf diesen Button

Hinweis: Mit **ZURÜCK** kommen sie wieder zur Startseite

## Auswahl

Auf dieser Seite können sie die einzelnen Elemente des Formulars zusammenstellen.



**TEXTFELD:** Für kurzen Text oder einzelne Wörter geeignet. Text Namen eingeben. Zum Beispiel: Strasse/Konfession/Zivilstand. Bestimmen Sie, ob es ein Pflichtfeld sein muss; Maximum Anzahl Zeichen bestimmen (mehr dazu auf Seite 6 unter Text).

**TEXT:** Dieses Feld eignet sich gut für einen längeren Text. Zum Beispiel für Mitteilungen. Sie können dem Textfeld einen Namen geben; bestimmen, ob es ein Pflichtfeld sein soll; Maximum Anzahl Zeichen bestimmen und mit/ohne PLZ (mehr dazu auf Seite 7 unter Textfeld).

**E-MAIL:** Spezielles Feld für E-Mail-Eingabe. Einen Namen geben und bestimmen ob es ein Pflichtfeld sein soll. (mehr dazu auf Seite 8 unter E-mail.)

**RADIOBOX:** Hier können Sie Auswahlfelder erstellen, zum Beispiel: Herr oder Frau/ ja oder nein – statt mit Text können Sie mit einem Klick eine Auswahl generieren. (mehr dazu auf Seite 9–10 unter Radiobox).

**CHECKBOX:** Wie Radiobox, nur können Sie hier mehrere Möglichkeiten auswählen. Zum Beispiel für eine Auswertung. Bezeichnung für den Checkboxnamen geben; mehrere Check-boxnamen geben und erstellen. (mehr dazu auf Seite 11–12 unter Checkbox.)

**DATUM:** Feld für Datumeingabe; Bezeichnung für Datumname; von – bis Eingaben; bestimmen ob es ein Pflichtfeld wird. (mehr dazu auf Seite 13 unter Datum).

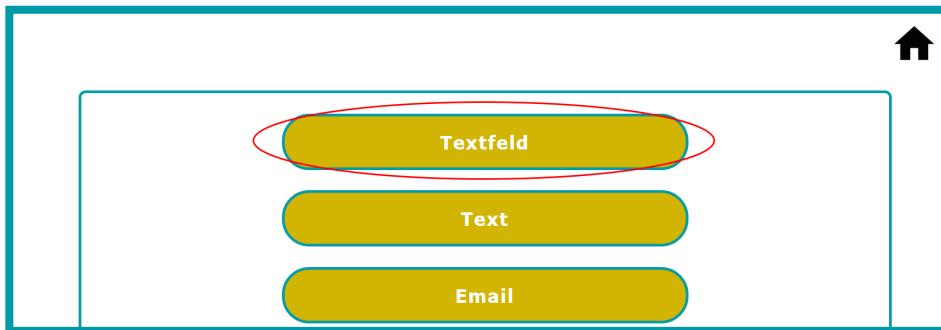
**NUMMER:** Spezielles Feld für Zahleneingaben. Bezeichnung für die Nummernamen angeben. Zum Beispiel: Menge/ Hausnummer/Betrag. Bestimmen ob es ein Pflichtfeld sein soll. Min./max. Zeichen bestimmen (mehr dazu auf Seite 14 unter Nummer.)

**AUSWAHLLISTE:** Hier erstellen Sie Menüleisten; der Auswahlliste einen Namen geben; Wahllisten erstellen. Zum Beispiel: Anmeldung, Bestellung, Anfrage (mehr dazu auf Seite 15–16 unter Auswahlliste.)

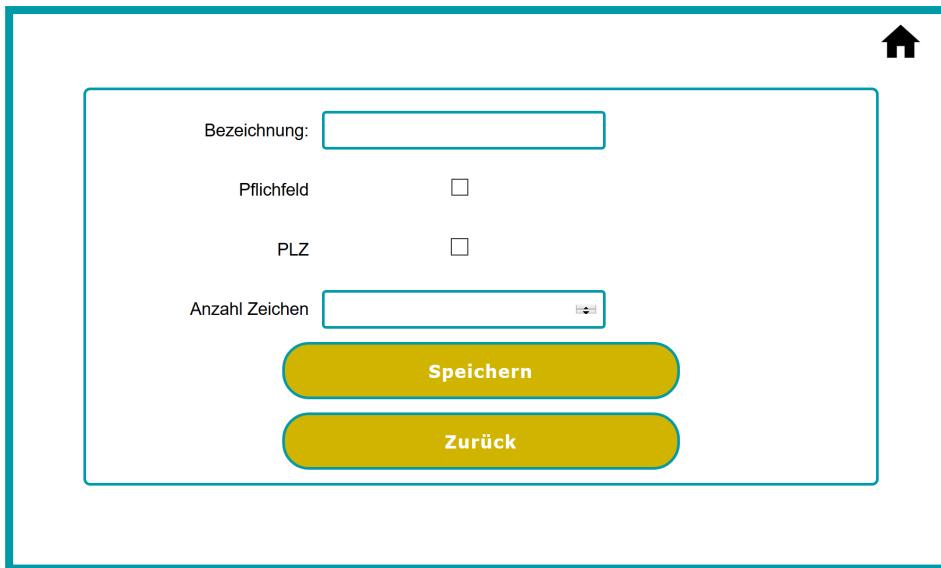
**DATEI LADEN:** Hier können Sie Bild- und Textdateien hochladen und einfügen. Sie bestimmen die maximale Dateigrösse. (mehr dazu auf Seite 17 unter Auswahlliste.)

## Textfeld erstellen

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - TEXTFELD



Klicken auf TEXTFELD

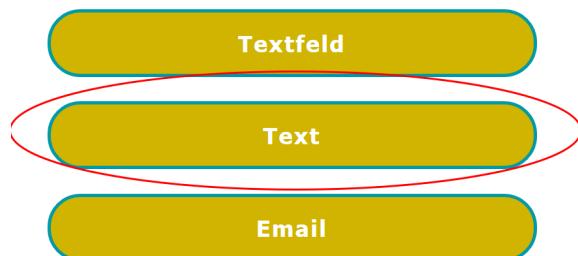


- Geben Sie eine Bezeichnung für das Textfeld ein. (Zum Beispiel Name/Vorname)
- Wählen Sie, ob das Textfeld ein Pflichtfeld sein muss oder nicht.
- Mit PLZ bestimmen Sie ob das Textfeld eine Postleitzahl beinhalten soll.
- Geben Sie die max. Anzahl Zeichen ein.
- Nach dem **SPEICHERN** gelangen Sie automatisch wieder zur Auswahlseite zurück.
- Mit dem Button **ZURÜCK** kehren Sie zur Auswahlseite zurück ohne zu speichern.

Wie Sie hier das Erstellte bearbeiten und/oder löschen können siehe Seite 18

## Text

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - **TEXT**



Klicken auf **TEXT5**

A screenshot of a form configuration interface. The interface has a blue header bar with a house icon. Below it is a white form area with a teal border. Inside the form area, there are several input fields and buttons:

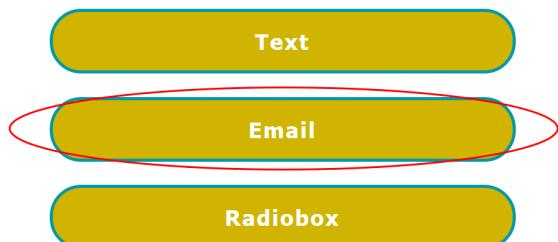
- A text input field labeled "Bezeichnung:" with a placeholder box.
- A checkbox labeled "Pflichtfeld" with an unchecked state.
- A text input field labeled "Max Zeichen" with a placeholder box and a right-pointing arrow icon.
- A yellow button labeled "Speichern".
- A yellow button labeled "Zurück".

- Geben Sie eine Bezeichnung für den Text ein. (Zum Beispiel: Mitteilung)
- Wählen Sie, ob das Textfeld ein Pflichtfeld sein muss oder nicht.
- Geben Sie die max. Anzahl Zeichen ein.
- Nach dem **SPEICHERN** gelangen Sie automatisch wieder zur Auswahlseite zurück.
- Mit dem Button **ZURÜCK** kehren Sie zur Auswahlseite zurück ohne zu speichern.

Wie Sie hier das Erstellte bearbeiten und/oder löschen können siehe Seite 18

## Email

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - **EMAIL**



Klicken auf **EMAIL**

Bezeichnung:

Pflichtfeld

**Speichern**

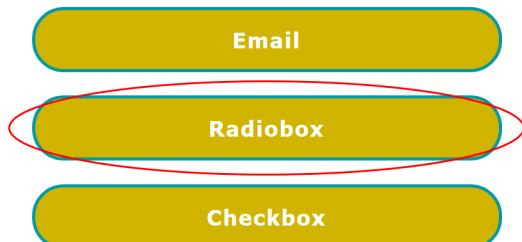
**Zurück**

- Geben Sie eine Bezeichnung für das E-Mailfeld ein. (Zum Beispiel: Email)
- Wählen Sie, ob das E-Mailfeld ein Pflichtfeld sein muss oder nicht.
- Geben Sie die max. Anzahl Zeichen ein.
- Nach dem **SPEICHERN** gelangen Sie automatisch wieder zur Auswahlseite zurück.
- Mit dem Button **ZURÜCK** kehren Sie zur Auswahlseite zurück ohne zu speichern.

Wie Sie hier das Erstellte bearbeiten und/oder löschen können siehe Seite 18

## Radiobox

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - **RADIOBOX**



Klicken auf **RADIOBOX**

The screenshot shows a form creation interface. At the top right is a house icon. Inside the main area, there is a text input field labeled 'Bezeichnung:' with the placeholder 'Anrede'. Below it are two yellow rounded rectangular buttons labeled 'Speichern' and 'Zurück'.

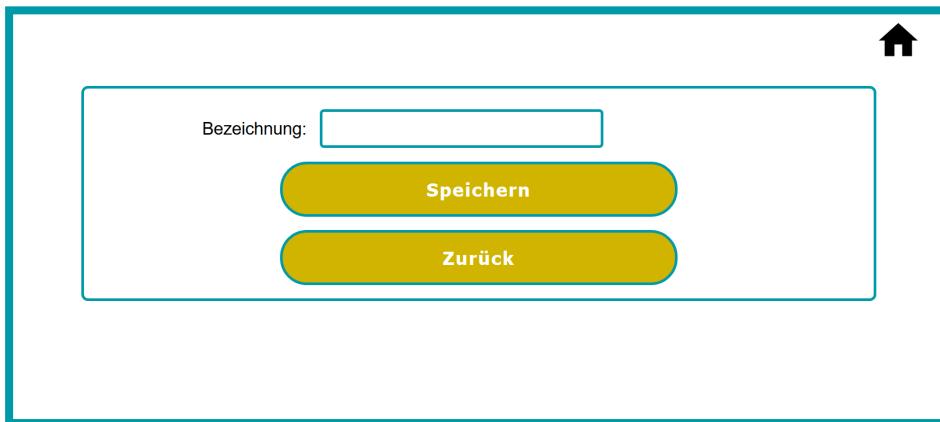
- Geben Sie einen Namen für die Radiobox ein. (Zum Beispiel Anrede)
- **SPEICHERN**, sie bleiben auf dieser Seite
- Button **RADIOBOX ERSTELLEN** erscheint

Hinweis: Mit dem Button **ZURÜCK** kehren Sie zur Auswahl zurück ohne Erstellung und Speicherung.

The screenshot shows the same form creation interface as before, but now with a fourth yellow rounded rectangular button at the bottom labeled 'Radiobox Erstellen'. The other elements ('Bezeichnung:', 'Speichern', 'Zurück') remain the same.

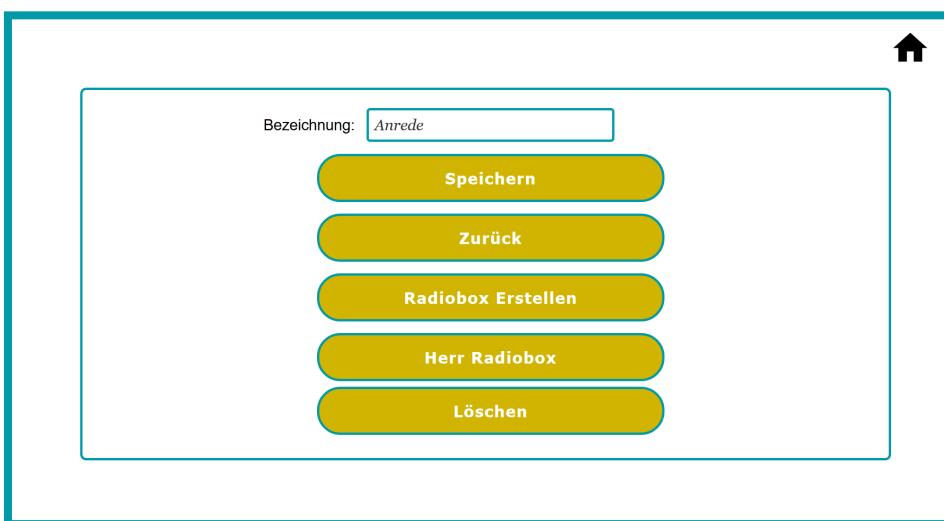
Klicken auf **RADIOBOX ERSTELLEN**

## Radiobox



Geben Sie einen Namen für das **Radioboxfeld** ein. (Zum Beispiel: Herr)  
**SPEICHERN**, Sie kommen wieder auf die Radiobox-Seite zurück

Hinweis: Mit dem Button **ZURÜCK** kehren Sie zur Radiobox-Seite zurück ohne Speicherung.



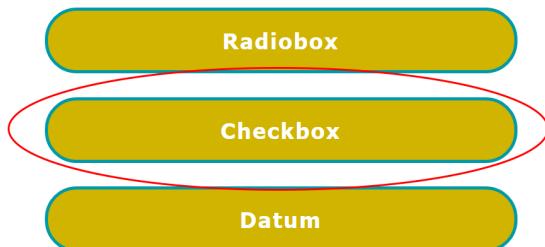
Auf der Seite erscheint nun der Name Ihres erstellten Radiobox-Feldes. (Beispiel: Herr).  
Nun können Sie mit Klick auf den Button **RADIOBOX ERSTELLEN** zusätzliche Radiobox-Felder erstellen, zum Beispiel: Frau. Speichern Sie die erstellten Radiobox-Namen ab.

- Mit dem Button **LÖSCHEN** werden die gespeicherten Felder gelöscht.
- Mit dem Button **ZURÜCK** kehren Sie zur Auswahl zurück, die Eingaben bleiben gespeichert.

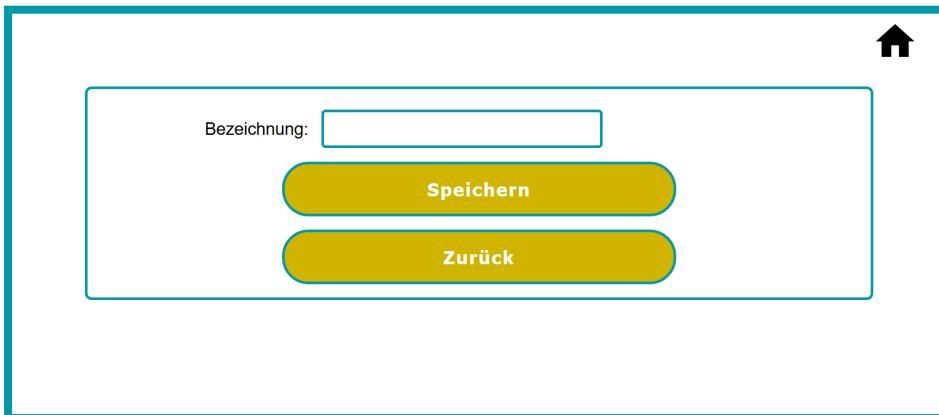
Wie Sie hier das Erstellte bearbeiten können siehe Seite 18

## Checkbox

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - **CHECKBOX**

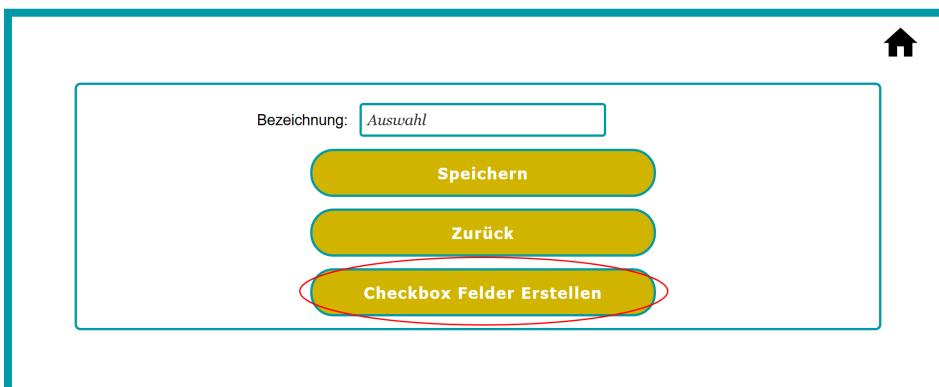


Klicken auf **CHECKBOX**



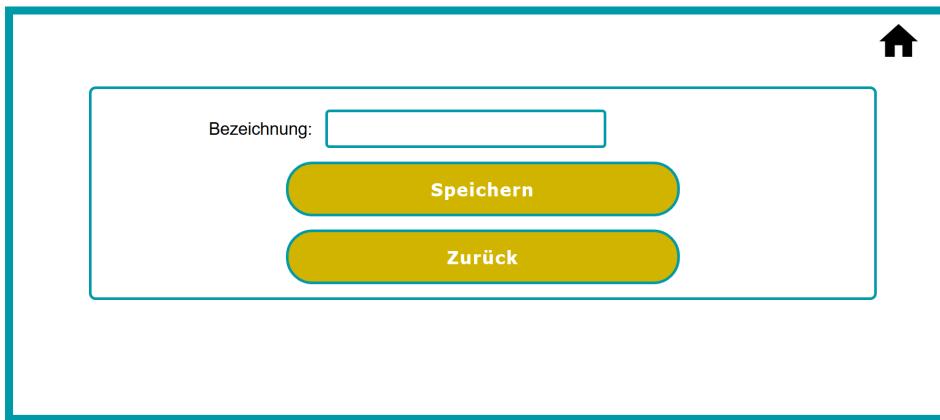
- Geben Sie einen Namen für die Checkbox ein. (Zum Beispiel Auswahl)
- **SPEICHERN**, sie bleiben auf dieser Seite
- Button **CHECKBOX FELDER ERSTELLEN** erscheint

Hinweis: Mit dem Button **ZURÜCK** kehren Sie zur Auswahl zurück ohne Erstellung und Speicherung.



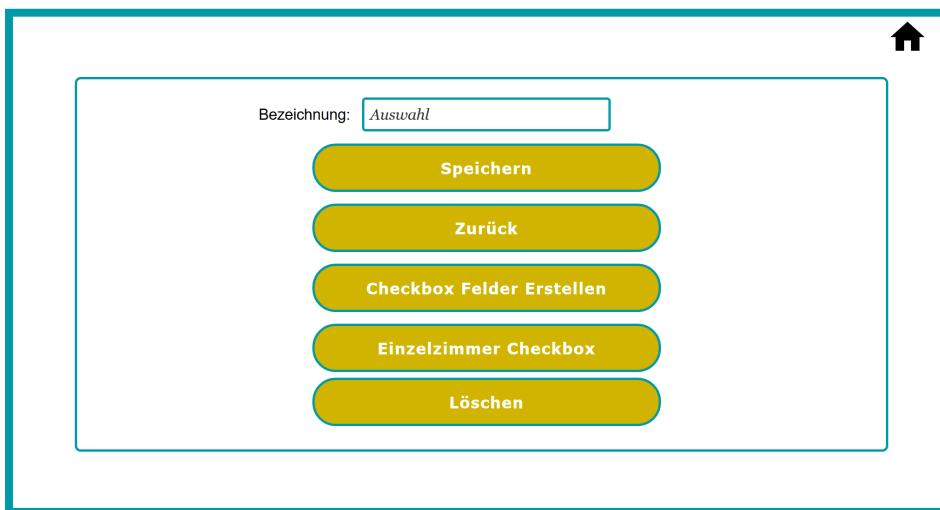
Klicken auf **CHECKBOX FELDER ERSTELLEN**

## Checkbox



- Geben Sie einen Namen für das Checkboxfeld ein. (Zum Beispiel: Einzelzimmer)
- **SPEICHERN**, Sie kommen wieder auf die Checkbox-Seite zurück

Hinweis: Mit dem Button **ZURÜCK** kehren Sie wieder zur Checkbox-Seite zurück ohne Speicherung.



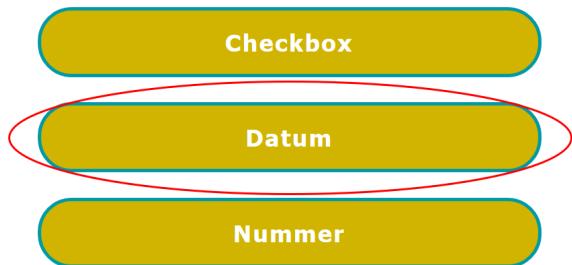
Auf der Seite erscheint nun der Name Ihres erstellten Checkbox-Feldes. (Beispiel: Einzelzimmer). Nun können Sie mit einem Klick auf den Button **CHECKBOX FELDER ERSTELLEN** zusätzliche Checkbox-Felder erstellen, zum Beispiel: Doppelzimmer, Meersicht usw). Speichern Sie die erstellten Checkbox-Namen ab.

- Mit dem Button **LÖSCHEN** werden die gespeicherten Felder gelöscht.
- Mit dem Button **ZURÜCK** kehren Sie zur Auswahl zurück, die Eingaben bleiben gespeichert.

Wie Sie hier das Erstellte bearbeiten können siehe Seite 18

## Datum

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - DATUM



Klicken auf DATUM

The screenshot shows a form configuration window with a teal border. At the top right is a house icon. Inside, there's a 'Bezeichnung:' input field with a placeholder '(tt.mm.jjjj)'. Below it is a checkbox labeled 'Pflichtfeld' with an unchecked square. There are two date input fields: 'Datum von:' and 'Datum bis:', both with the placeholder '(tt.mm.jjjj)'. At the bottom are two buttons: a yellow 'Speichern' button and a blue 'Zurück' button.

- Geben Sie eine Bezeichnung für das Datumfeld ein. (Zum Beispiel tt.mm.jj)
- Wählen Sie, ob das Datumfeld ein Pflichtfeld sein muss oder nicht.
- Mit den Feldern von/bis können Sie das Datum einschränken
- **SPEICHERN**, Ihre Eingaben bleiben gespeichert

Hinweis: Mit dem Button **ZURÜCK** kehren Sie wieder zur Auswahl zurück ohne Erstellung und Speicherung.

Wie Sie hier das Erstellte bearbeiten können siehe Seite 18

## Nummer

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL - **NUMMER**



Klicken auf **NUMMER**

Bezeichnung:

Pflichtfeld

Min. wert

Max. wert

**Speichern**

**Zurück**

- Geben Sie eine Bezeichnung für das Nummernfeld ein. (Zum Beispiel: Kapitel)
- Wählen Sie, ob das Nummernfeld ein Pflichtfeld sein muss oder nicht.
- Mit den Feldern min. Wert/max. Wert können Sie den Wert einschränken
- **SPEICHERN**, Ihre Eingaben bleiben gespeichert und Sie kehren automatisch auf die Auswahlseite zurück

Hinweis: Mit dem Button **ZURÜCK** kehren Sie wieder zur Auswahlseite zurück ohne Erstellung und Speicherung.

Wie Sie hier das Erstellte bearbeiten können siehe Seite 18

## Auswahlliste

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL – **AUSWAHLLISTE**



Klicken auf **AUSWAHLLISTE**

This screenshot shows the 'Auswahlliste' creation form. It includes a text input field labeled 'Bezeichnung:' with the placeholder 'Kursorte', a 'Speichern' (Save) button, and a 'Zurück' (Back) button. The entire form is enclosed in a light blue border.

- Geben Sie einen Namen für die Auswahlliste ein. (Zum Beispiel: KURSORTE)
- **SPEICHERN**, Sie kommen wieder auf die Auswahl-Seite zurück

Hinweis: Mit dem Button **ZURÜCK** kehren Sie wieder zur Auswahl-Seite zurück ohne Speicherung.

This screenshot shows the 'Auswahlliste' creation form after entering a name. It now displays 'Bezeichnung: Kursorte' in the input field. The 'Speichern' and 'Zurück' buttons remain, while the 'Auswahl Erstellen' (Create Selection) button at the bottom is highlighted with a red oval.

Klicken auf **AUSWAHL ERSTELLEN**

## Auswahlliste

The screenshot shows a user interface for creating a new selection field. At the top right is a house icon. Below it is a text input field labeled "Bezeichnung:" containing the placeholder text "Kursort". Underneath the input field are two yellow rounded rectangular buttons: "Speichern" and "Zurück".

- Geben Sie einen Namen für das Auswahlfeld ein. (Zum Beispiel: Basel)
- **SPEICHERN**, Sie kommen wieder auf die Auswahlliste-Seite zurück

Hinweis: Mit dem Button **ZURÜCK** kehren Sie zur Auswahlliste-Seite zurück ohne Speicherung.

The screenshot shows a user interface for managing selection fields. At the top right is a house icon. Below it is a list of five yellow rounded rectangular buttons:

- Bezeichnung: *Kursort*
- Speichern
- Zurück
- Auswahl Erstellen
- Basel Auswahlliste
- Löschen

Auf der Seite erscheint nun der Name Ihres erstellten Auswahl-Feldes. (Beispiel: Basel). Nun können Sie mit Klick auf den Button **AUSWAHL ERSTELLEN** zusätzliche Auswahl-Felder erstellen, zum Beispiel: Zürich. Speichern Sie die erstellten Auswahl-Namen ab.

- Mit dem Button **LÖSCHEN** werden die gespeicherten Felder gelöscht.
- Mit dem Button **ZURÜCK** kehren Sie wieder zur Auswahlseite zurück, die Eingaben bleiben gespeichert.

Wie Sie hier das Erstellte bearbeiten können siehe Seite 18

## Datei hochladen

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL – DATEI HOCHLADEN



Klicken auf DATEI HOCHLADEN

Bezeichnung:

Pflichtfeld

**Speichern**

**Zurück**

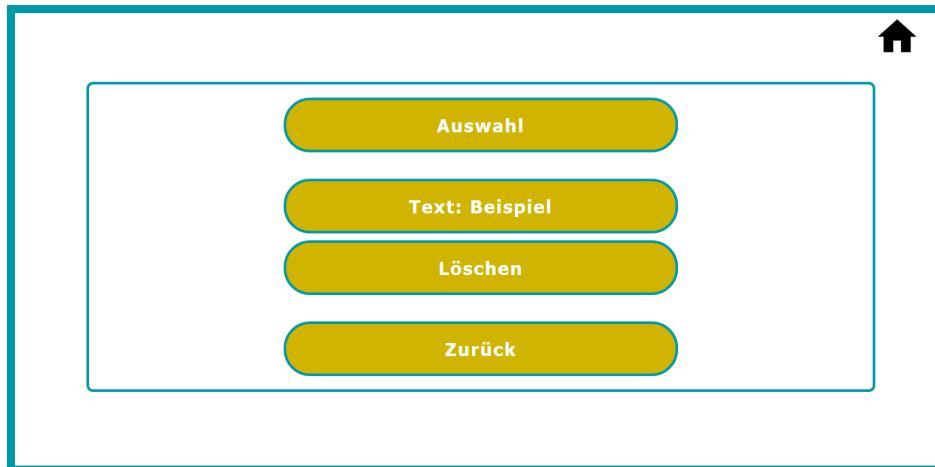
- Geben Sie eine Bezeichnung für das Feld ein. (Zum Beispiel: Anhang Datei)
- Wählen Sie, ob das Feld ein Pflichtfeld sein muss oder nicht.
- Geben Sie den maximalen Grösse der Datei an.
- **SPEICHERN**, Ihre Eingaben bleiben gespeichert und Sie kehren automatisch auf die Auswahlseite zurück

Hinweis: Mit dem Button **ZURÜCK** kehren Sie zur Auswahl zurück ohne Erstellung und Speicherung.

Wie Sie hier das Erstellte bearbeiten können siehe Seite 18

## Einzelne Elemente löschen oder Bearbeiten

Pfad: FORMULAR ERSTELLEN – BEARBEITEN – AUSWAHL – **BEARBEITEN**



- Klicken Sie auf das zu korrigierende Button
- Korrigieren und/oder ändern Sie den Text
- **SPEICHERN**
- Mit dem Button **LÖSCHEN** werden die gespeicherten Felder gelöscht.

## Quellenverzeichnis

### ABBILDUNGEN

Abb. 1: Skizze	Seite 22
Abb. 2: Variante 1A	Seite 23
Abb. 3: Variante 1B	Seite 23
Abb. 4: Variante 2A	Seite 24
Abb. 5: Variante 2B	Seite 24
Abb. 6: Variante 3A	Seite 25
Abb. 7: Variante 3B	Seite 25
Abb. 8: Endprodukt	Seite 27
Abb. 9: Haussymbol	Seite 28
Abb. 10: Formular erstellen	Seite 28
Abb. 11: Formularübersicht	Seite 29
Abb. 12: Auswahl aller Inputs	Seite 30
Abb. 13: Input	Seite 31
Abb. 14: Eingabe des Textfeldes	Seite 31
Abb. 15: Eingabe des Textes	Seite 32
Abb. 16: Eingabe des Feldes Email	Seite 32
Abb. 17: Eingabe der Radiobox	Seite 32
Abb. 18: Bezeichnung der Radiobox	Seite 33
Abb. 19: Hinzufügen der Radiobox	Seite 33
Abb. 20: Eingabe der Checkbox	Seite 34
Abb. 21: Bezeichnung der Checkbox	Seite 34
Abb. 22: Hinzufügen der Checkbox	Seite 35
Abb. 23: Eingabe des Feldes Datum	Seite 35
Abb. 24: Eingabe der Feldes Nummer	Seite 36
Abb. 25: Eingabe der Auswahlliste	Seite 36
Abb. 26: Bezeichnung der Auswahlliste	Seite 37
Abb. 27: Hinzufügen eines weiteren Elementes	Seite 37
Abb. 28: Eingabe des Feldes Datei hochladen	Seite 38
Abb. 29: Ansichtsbeispiel der Buttons	Seite 39
Abb. 30: Ansichtsbeispiel der Textfelder	Seite 39

## QUELLEN

PHP Programmierungshilfe:

<http://php.net>

Programmierungshilfe CSS:

<https://wiki.selfhtml.org/wiki/Schnell-Index/CSS>

E-Mail versenden:

<https://www.sanwebe.com/2015/12/php-mail-with-multiple-attachments>

Home Button:

[http://cdn.onlinewebfonts.com/svg/img\\_5503.svg](http://cdn.onlinewebfonts.com/svg/img_5503.svg)

savearray in Funktion.php:

Firma asperger-ag

## TABELLEN

Tab. 1: Zeitplanung	Seite 8
Tab. 2: Zielsetzung und Abweichung	Seite 20
Tab. 3: Kontrolle Formulargenerator	Seite 41
Tab. 4: Auftragstest	Seite 41
Tab. 5: Kontrolle Anleitung	Seite 42
Tab. 6: Kontrolle Dokumentation	Seite 42
Tab. 7: Qualitätssicherung	Seite 43

## Glossar

PHP	Hypertext Preprocessor“, ursprünglich „Personal Home Page Tools“
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
CMS	Content-Management-System
FTP	File Transfer Protocol

## Anhang

### FIRMENSTANDARDS

#### PHP Richtlinien

- Sinnvolle Benennung der Variablen, Array (Beispiel \$counter statt \$x)
- Nach Vergleichsoperatoren ein Abstand
- Schliessende und öffnende Klammern auf einer neuen Zeile
- Einrücken des Codes
- Einfache Anführungszeichen verwenden
- Kommentare falls nötig
- Öffnen und Schliessen von PHP auf einer neuen Zeilen

```
<?php  
// Beispiel  
if ($counter > 2)  
{  
    if ($counter === 77)  
    {  
        echo 'Hallo';  
    }  
}  
?>
```

## **Namenskonvention für Dateien**

- Alles klein geschrieben
- Keine Sonderzeichen
- Für Dateien wo man direkt im Web verwendet:  
Statt Leerzeichen – verwenden
- Für alles andere: statt Leerzeichen \_ verwenden
- Beschreibender Name Verwenden asperger-logo.jpg statt  
image123.jpg
- Umlaute ä,ö,ü vermeiden, stattdessen ae,oe,ue verwenden  
baer.jpg statt Bär.JPG

## CODE

### index.php

```
<?php
    include('function.php');
    include('header.php');
    if (isset( $_POST[,formedit' ] )) {
        include('formview.php');
        exit();
    }
    if ( isset( $_POST[,save' ] ))
    {
        $form[] = array(id'=>0, ,type'=>'form', ,form'=>$_POST[,formname'], ,email'=>$_POST[,email'], ,max'=>$_POST[,max']);
        if (isset( $_POST[,formname' ] ))
        {
            if (isset( $_POST[,form'] ) && !isset( $_POST[,copy'] ) )
            {
                $old=('../form/'.$_POST[,form'].php';
                $new='..form/'.$_POST[,formname'].php';
                rename($old , $new);
            }
            savearray($form, ,form', ..../form/'.$_POST[,formname'].php');
        }
        header(Location: index.php');
    }
    elseif ( isset( $_POST[,delete' ] ))
    {
        if (file_exists(..../form/'.$_POST[,delete'].php')){
            unlink(..../form.'/'.$_POST[,delete'].php');
        }
        header(Location: index.php');
    }
    elseif ( isset( $_POST[,new' ] ) || isset( $_POST[,form' ] ) )
    {
        include('form.php');
    }
    else
    {
?>
        <div class="linie">
            <form method="post">
                <fieldset>
                    <button name="new" type="submit">Formular Erstellen</button>
                </fieldset>
            <?php
                $path = ..../form/';
                $files = scandir($path);
                foreach ($files as $file) {
                    $filename = pathinfo($path."/".$file);
                    if ( $file != ..' && $file != ...' && !is_dir($path . '/' . $file))
                    {
                        echo ,<br /><h2>$filename[filename] .</h2>;
                    }
                }
            <?>
```

```

</form>
<form method="post" action="output.php?formname=<?php
echo $filename[,filename'];
?>" target="_blank">
    <button name="output" type="submit" value="output.php?formname=<?php
echo $filename[,filename'];
?>" target="_blank">
        Anschauen
    </button>
</form>
<form method="post" action="formview.php">
    <button name="formname" type="submit" value="<?php
echo $filename[,filename'];
?>">
        Bearbeiten
    </button>
</form>
<form method="post">
    <button name="form" type="submit" value="<?php
echo $filename[,filename'];
?>">
        Einstellungen
    </button>
</form>
<form method="post">
    <button name="form" type="submit" value="<?php
echo $filename[,filename'];
?>">
        Kopieren
    </button>
    <input type="hidden" name="copy" value="copy">
</form>
<form method="post">
    <button name="delete" type="submit" value="<?php echo
$filename[,filename'];
?>" onClick="return confirm(<?php
echo $filename[,filename'];
?> löschen?')">
        Löschen
    </button>
</form>
<?php
}
?
?>
</div>

<?php
}
include('footer.php');
?>

```

```

form.php
<?php
exit();
$formname = ,';
$email = ,';
$reply = ,';
$subject = ,';
if (isset( $_POST[,form'] ))
{
    if ( file_exists(.../form/'. $_POST[,form']. '.php' ) )
    {
        include .../form/'. $_POST[,form']. '.php';
        $form=unserialize($form);
        foreach ($form as $output) {
            if ($output[,id'] == 0){
                $formname=$output[,form'];
                $email=$output[,email'];
                $reply=$output[,reply'];
                $subject=$output[,subject'];
            }
        }
    }
}

if (isset( $_POST[,formname'] ))
{
    $formname = $_POST[,formname'];
}
if (isset( $_POST[,email'] ))
{
    $email = $_POST[,email'];
}
if (isset( $_POST[,reply'] ))
{
    $reply = $_POST[,reply'];
}
if (isset( $_POST[,subject'] ))
{
    $subject = $_POST[,subject'];
}
?>
<div class="linie">
<form method="post">
<fieldset>
<div class="form">
    <label for="formname">Formularname<sup>*</sup></label>
    <input type="text" name="formname" id="formname" value="<?php echo $formname; ?>" required />
</div>
<div class="email">
    <label for="email">E-Mail<sup>*</sup></label>
    <input type="email" name="email" id="email" value="<?php echo $email; ?>" required />
</div>
<div class="reply">
    <label for="reply">Antwortadresse<sup>*</sup></label>
    <input type="email" name="reply" id="reply" value="<?php echo $reply; ?>" required />
</div>
<div class="subject">

```

```

<label for="subject">Betreff<sup>*</sup></label>
<input type="text" name="subject" id="subject" value=<?php echo $subject; ?>" required />
</div>
<?php
if (isset( $_POST[,form'] ))
{
?>
<input type="hidden" name="form" value=<?php echo $_POST[,form']; ?>" required />
<?php
}
?>
</fieldset>
<button name="save" type="submit">Weiter</button>
</form>
<form method="post" action="index.php">
<button name="submit" type="submit" value="index.php">Zurück</button>
</form>
</div>

```

### funktion.php

```

<?php
/*function savearray von cms kopiert (Betriebsintern)*/
function savearray ($array, $variable, $file){
    $content = ,<?php ..‘.$‘.“.$variable.“=“serialize($array).“; ?>“;
    $fp = fopen($file,’w+’);
    fwrite($fp, $content);
    fclose($fp);
}

function deletearray ($array, $id){
    foreach ($array as $subkey=>$subarray)
    {
        if ($subarray[‘id’] == $id )
        {
            unset($array[$subkey]);
        }
    }
    return $array;
}
?>

```

### header.php

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link href= „/css/design/style.css“ rel="stylesheet">
</head>
<body>
    <div class="wrapper">
        <a href="index.php"><img src=„/css/images/home.png“ wight="30px" height="30px" type="TRUE" alt="+"></a>
        <div class="main" role="main">
footer.php
    </div>
</div>
</body>
</html>

```

```

formview.php
<?php
include('function.php');
include('header.php');
if ( !isset( $_POST[,formname'] ) )
{
    header('Location: index.php');
    exit();
}

include .../form/'. $_POST[,formname']. ' .php';
$form = unserialize($form);
$max = max( array_column($form, ,id') );
if ( isset($_POST[,type']) )
{
    $type = $_POST[,type'];
}
else
{
    $_POST[,type'] = ,;
    $type = ,;
}

if ( isset($_POST[,submit']) )
{
    $button = $_POST[,submit'];
}
else{
    $button = ,;
}

if ( isset( $_POST[,plus'] ) )
{
    if ( $_POST[,type'] !== ,check' && $_POST[,type'] !== ,radio' && $_POST[,type'] !== ,option' )
    {
        include('input.php');
        exit();
    }
}
if ( !isset( $_POST[,required'] ) )
{
    $_POST[,required'] = ,;
}
if ( !isset( $_POST[,plz'] ) )
{
    $_POST[,plz'] = ,;
}
if ( !isset( $_POST[,fk_id'] ) )
{
    $_POST[,fk_id'] = ,;
}
switch ($type)
{
    case ,text':
        switch ($button)

```

```

{
    case ,save':
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] )
                {
                    $output[,required'] = $_POST[,required'];
                    $output[,designation'] = $_POST[,designation'];
                    $output[,max'] = $_POST[,max'];
                    $output[,plz'] = $_POST[,plz'];
                }
            }
        }
    else
    {
        $id = $max+1;
        $form[] = array(id'=>$id, ,type'=>'text', ,required'=>$_POST[,required'], ,designation'=>$_POST[,designation'], ,max'=>$_POST[,max'], ,plz'=>$_POST[,plz']);
    }
    savearray($form, ,form', ../form/'. $_POST[,formname'].php');
    break;
    case ,delete':
        savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
        break;
    default:
        include,(inputs/text.php');
        exit();
        break;
    }
    break;

case ,email':
    switch ($button)
    {
        case ,save':
            if ( isset( $_POST[,id'] ) )
            {
                foreach ($form as &$output)
                {
                    if ( $_POST[,id'] == $output[,id'] )
                    {
                        $output[,required'] = $_POST[,required'];
                        $output[,designation'] = $_POST[,designation'];
                    }
                }
            }
        else
        {
            $id = $max+1;
            $form[] = array(id'=>$id, ,type'=>'email', ,required'=>$_POST[,required'], ,designation'=>$_POST[,designation']);
        }
        savearray($form, ,form', ../form/'. $_POST[,formname'].php');
        break;
        case ,delete':
            savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
            break;
        default:
            include,(inputs/email.php');
            exit();
            break;
    }
}

```

```

        }
        break;

case ,number':

switch ($button)
{
    case ,save':
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] )
                {
                    $output[,required'] = $_POST[,required'];
                    $output[,designation'] = $_POST[,designation'];
                    $output[,min'] = $_POST[,min'];
                    $output[,max'] = $_POST[,max'];
                }
            }
        }
        else
        {
            $id = $max+1;
            $form[] = array(id'=>$id, ,type'=>'number', ,required'=>$_POST[,required'], ,designation'=>$_POST[,designation'], ,min'=>$_POST[,min'], ,max'=>$_POST[,max']);
        }
        savearray($form, ,form', ../form/'.$_POST[,formname'].'.php');
        break;
    case ,delete':
        savearray( deletearray($form,$_POST[,id']) , ,form', ../form.'/'.$_POST[,formname'].'.php');
        break;
    default:
        include,(inputs/number.php');
        exit();
        break;
}
break;

case ,textarea':

switch ($button)
{
    case ,save':
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] )
                {
                    $output[,required'] = $_POST[,required'];
                    $output[,designation'] = $_POST[,designation'];
                    $output[,max'] = $_POST[,max'];
                }
            }
        }
        else
        {
            $id = $max+1;
            $form[] = array(id'=>$id, ,type'=>'textarea', ,required'=>$_POST[,required'], ,designation'=>$_POST[,designation'], ,max'=>$_POST[,max']);
        }
        savearray($form, ,form', ../form.'/'.$_POST[,formname'].'.php');
        break;
}

```

```

case ,delete':
    savearray( deletearray($form,$_POST[,id']) , ,form' , ../form/'.$_POST[,formname'].php');
    break;
default:
    include(,inputs/textarea.php');
    exit();
    break;
}
break;

case ,date':

switch ($button)
{
    case ,save':
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] )
                {
                    $output[,required'] = $_POST[,required'];
                    $output[,designation'] = $_POST[,designation'];
                    $output[,min'] = $_POST[,min'];
                    $output[,max'] = $_POST[,max'];
                }
            }
        }
        else
        {
            $id = $max+1;
            $form[] = array(id'=>$id, ,type'=>'date' , ,required'=>$_POST[,required'], ,designation'=>$_POST[,designation'],
            ,min'=>$_POST[,min'], ,max'=>$_POST[,max']);
        }
        savearray($form, ,form' , ../form/'.$_POST[,formname'].php');
        break;
    case ,delete':
        savearray( deletearray($form,$_POST[,id']) , ,form' , ../form/'.$_POST[,formname'].php');
        break;
    default:
        include(,inputs/date.php');
        exit();
        break;
}
break;

case ,fileupload':

switch ($button)
{
    case ,save':
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] )
                {
                    $output[,required'] = $_POST[,required'];
                    $output[,designation'] = $_POST[,designation'];
                }
            }
        }
        else
        {
            $id = $max+1;

```

```

$form[] = array(,id'=>$id, ,type'=>'fileupload', ,required'=>$_POST[,required'], ,designation'=>$_POST[,designation']);
}
savearray($form, ,form', ../form/'. $_POST[,formname'].php');
break;
case ,delete':
savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
break;
default:
include(,inputs/fileupload.php');
exit();
break;
}
break;

case ,select':
switch ($button)
{
case ,save':
$id = 0;
if ( isset( $_POST[,id'] ) )
{
foreach ($form as &$output)
{
if ( $_POST[,id'] == $output[,id'] )
{
$output[,designation'] = $_POST[,designation'];
$id = $output[,id'];
}
}
}
else
{
$id = $max + 1;
$form[] = array(,id'=>$id, ,type'=>'select', ,designation'=>$_POST[,designation']);
}
savearray($form, ,form', ../form/'. $_POST[,formname'].php');
$newid = $id;
$children = 1;
include(,inputs/select.php');
exit();
break;
case ,delete':
savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
$children = 1;
break;
default:
include(,inputs/select.php');
exit();
break;
}
break;

case ,option':
switch ($button)
{
case ,save':
$id = 0;
if ( isset( $_POST[,id'] ) )
{
foreach ($form as &$output)
{

```

```

        if ( $_POST[,id'] == $output[,id'] && $_POST[,fk_id'] == $output[,fk_id'] ){
            $output[,designation']= $_POST[,designation2'];
            $id = $output[,fk_id'];
        }
    }
}
else
{
    $id = $max+1;
$form[] = array(id'=>$id, ,fk_id'=>$_POST[,fk_id'], ,type'=>'option', ,designation'=>$_POST[,designation2']);
    $id = $_POST[,fk_id'];
}
savearray($form, ,form', ../form/'. $_POST[,formname'].php');
include ../form/'. $_POST[,formname'].php';
$form = unserialize($form);
$newid = $id;
$_POST[,id'] = $id;
include(,inputs/select.php');
exit();
break;
case ,delete':
    savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
    $newid = $_POST[,fk_id'];
    $_POST[,id'] = $_POST[,fk_id'];
    include ../form/'. $_POST[,formname'].php';
    $form = unserialize($form);
    include(,inputs/select.php');
    exit();
    break;
default:
    include(,inputs/option.php');
    exit();
    break;
}
break;

case ,radiobox':
switch ($button)
{
    case ,save':
        $id = 0;
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] ){
                    $output[,designation'] = $_POST[,designation'];
                    $id = $output[,id'];
                }
            }
        }
    else
    {
        $id = $max+1;
        $form[] = array(id'=>$id, ,type'=>'radiobox', ,designation'=>$_POST[,designation']);
    }
    savearray($form, ,form', ../form/'. $_POST[,formname'].php');
    $newid = $id;
    include(,inputs/radiobox.php');
    exit();
    break;
}
case ,delete':
    savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');

```

```

        break;
    default:
        include(,inputs/radiobox.php');
        exit();
        break;
    }
    break;

case ,radio':

switch ($button)
{
    case ,save':
        $id = 0;
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] && $_POST[,fk_id'] == $output[,fk_id'] )
                {
                    $output[,designation'] = $_POST[,designation2'];
                    $id = $output[,fk_id'];
                }
            }
        }
        else
        {
            $id = $max+1;
            $form[] = array(id'=>$id ,fk_id'=>$_POST[,fk_id'], ,type'=>'radio', ,designation'=>$_POST[,designation2']);
            $id = $_POST[,fk_id'];
        }
        savearray($form, ,form', .../form/'.$_POST[,formname'].php');
        include .../form.'/'.$_POST[,formname'].php';
        $form = unserialize($form);
        $newid = $id;
        $_POST[,id'] = $id;
        include(,inputs/radiobox.php');
        exit();
        break;
    case ,delete':
        savearray( deletearray($form,$_POST[,id']) , ,form', .../form.'/'.$_POST[,formname'].php');
        $newid = $_POST[,fk_id'];
        $_POST[,id'] = $_POST[,fk_id'];
        include .../form.'/'.$_POST[,formname'].php';
        $form = unserialize($form);
        include(,inputs/radiobox.php');
        exit();
        break;
    default:
        include(,inputs/radio.php');
        exit();
        break;
}
break;

case ,:

switch ($button) {
    case ,save':
        $id = 0;
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {

```

```

if ( $_POST[,id'] == $output[,id'] )
{
    $output[,designation'] = $_POST[,designation'];
    $id = $output[,id'];
}
}

else
{
    $id = $max + 1;
    $form[] = array(id'=>$id, ,type'=>'checkbox', ,designation'=>$_POST[,designation']);
}
savearray($form, ,form', ../form/'. $_POST[,formname'].php');
$newid = $id;
include(,inputs/checkbox.php');
exit();
break;
case ,delete':
    savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
    break;
default:
    include(,inputs/checkbox.php');
    exit();
    break;
}
break;

case ,check':
switch ($button) {
    case ,save':
        $id = 0;
        if ( isset( $_POST[,id'] ) )
        {
            foreach ($form as &$output)
            {
                if ( $_POST[,id'] == $output[,id'] && $_POST[,fk_id'] == $output[,fk_id'] )
                {
                    $output[,designation']==$_POST[,designation2'];
                    $id = $output[,fk_id'];
                }
            }
        }
        else
        {
            $id = $max + 1;
            $form[] = array(id'=>$id, ,fk_id'=>$_POST[,fk_id'], ,type'=>'checkbox', ,designation'=>$_POST[,designation2']);
            $id = $_POST[,fk_id'];
        }
    savearray($form, ,form', ../form/'. $_POST[,formname'].php');
    include ../form/'. $_POST[,formname'].php';
    $form = unserialize($form);
    $newid = $id;
    $_POST[,id'] = $id;
    include(,inputs/checkbox.php');
    exit();
    break;
}
case ,delete':
    savearray( deletearray($form,$_POST[,id']) , ,form', ../form/'. $_POST[,formname'].php');
    $newid = $_POST[,fk_id'];
    $_POST[,id'] = $_POST[,fk_id'];
    include ../form/'. $_POST[,formname'].php';
    $form = unserialize($form);
    include(,inputs/checkbox.php');
}

```

```

        exit();
        break;
    default:
        include(,inputs/check.php');
        exit();
        break;
    }
    break;
}

include ../form/'. $_POST[,formname'].php';
$form = unserialize($form);
?>
<div class="linie">
<form method="post">
<button NAME="plus" type="submit">
    Auswahl
</button>
<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
</form>

<form method="post">
<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
</form>
<?php
foreach ($form as $output)
{
    switch ($output[,type'])
    {
        case ,number':
        ?>
            <form method="post">
                <button type="submit" name="submit">Nummer: <?php
                echo $output[,designation'];
                ?></button>
                <input type="hidden" name="id" value=<?php
                echo $output[,id'];
                ?>">
                <input type="hidden" name="type" value="number">
                <div class="delete">
                    <button type="submit" name="submit" value="delete">Löschen</button>
                    <input type="hidden" name="formname" value=<?php
                    echo $_POST[,formname'];
                    ?>">
                </div>
            </form><?php
        break;
    case ,email':
    ?>
        <form method="post">
            <button type="submit" name="submit">Email: <?php
            echo $output[,designation'];
            ?></button>
            <input type="hidden" name="id" value=<?php
            echo $output[,id'];
            ?>">
            <input type="hidden" name="type" value="email">
            <div class="delete">
                <button type="submit" name="submit" value="delete">Löschen</button>

```

```

<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
</div>
</form><?php
break;
case ,textarea':
?>
<form method="post">
<button type="submit" name="submit">Text: <?php
echo $output[,designation'];
?></button>
<input type="hidden" name="id" value=<?php
echo $output[,id'];
?>">
<input type="hidden" name="type" value="textarea">
<div class="delete">
<button type="submit" name="submit" value="delete">Löschen</button>
<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
</div>
</form><?php
break;
case ,text':
?>
<form method="post">
<button type="submit" name="submit">Textfeld: <?php
echo $output[,designation'];
?></button>
<input type="hidden" name="id" value=<?php
echo $output[,id'];
?>">
<input type="hidden" name="type" value="text">
<div class="delete">
<button type="submit" name="submit" value="delete">Löschen</button>
<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
</div>
</form><?php
break;
case ,date':
?>
<form method="post">
<button type="submit" name="submit">Datum: <?php
echo $output[,designation'];
?></button>
<input type="hidden" name="id" value=<?php
echo $output[,id'];
?>">
<input type="hidden" name="type" value="date">
<form method="post">
<div class="delete">
<button type="submit" name="submit" value="delete">Löschen</button>
<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
</div>
</form><?php
break;
case ,fileupload':
?>
<form method="post">

```

```

<button type="submit" name="submit">Datei hochladen: <?php
echo $output[,designation'];
?></button>
<input type="hidden" name="id" value="<?php
echo $output[,id];
?>">
<input type="hidden" name="type" value="fileupload">
<div class="delete">
    <button type="submit" name="submit" value="delete">Löschen</button>
    <input type="hidden" name="formname" value="<?php
echo $_POST[,formname]';
?>">
</div>
</form><?php
break;
case ,checkbox':
?>
<form method="post">
    <button type="submit" name="submit">Checkbox: <?php
echo $output[,designation'];
?></button>
    <input type="hidden" name="id" value="<?php
echo $output[,id];
?>">
    <input type="hidden" name="type" value="checkbox">
    <div class="delete">
        <button type="submit" name="submit" value="delete">Löschen</button>
        <input type="hidden" name="formname" value="<?php
echo $_POST[,formname]';
?>">
    </div>
</form><?php
break;
case ,select':
?>
<form method="post">
    <button type="submit" name="submit">Auswahlliste: <?php
echo $output[,designation'];
?></button>
    <input type="hidden" name="id" value="<?php
echo $output[,id];
?>">
    <input type="hidden" name="type" value="select">
    <div class="delete">
        <button type="submit" name="submit" value="delete">Löschen</button>
        <input type="hidden" name="formname" value="<?php
echo $_POST[,formname]';
?>">
    </div>
</form><?php
break;
case ,radiobox':
?>
<form method="post">
    <button type="submit" name="submit">Radiobox: <?php
echo $output[,designation'];
?></button>
    <input type="hidden" name="id" value="<?php
echo $output[,id];
?>">
    <input type="hidden" name="type" value="radiobox">
    <div class="delete">
        <button type="submit" name="submit" value="delete">Löschen</button>
        <input type="hidden" name="formname" value="<?php

```

```
echo $_POST[,formname'];
?>“>
</div>
</form>
<?php
break;
}
}
?>
<form method=“post“>
    <button name=“submit“ type=“submit“>Zurück</button>
</form>
</div>
<?php
include(,footer.php');
?>
```

```

input.php
<div class="linie">
  <form method="post">
    <div>
      <button type="submit" name="type" value="text">Textfeld</button>
    </div>
    <div>
      <button type="submit" name="type" value="textarea">Text</button>
    </div>
    <div>
      <button type="submit" name="type" value="email">Email</button>
    </div>
    <div>
      <button type="submit" name="type" value="radiobox">Radiobox</button>
    </div>
    <div>
      <button type="submit" name="type" value="checkbox">Checkbox</button>
    </div>
    <div>
      <button type="submit" name="type" value="date">Datum</button>
    </div>
    <div>
      <button type="submit" name="type" value="number">Nummer</button>
    </div>
    <div>
      <button type="submit" name="type" value="select">Auswahlliste</button>
    </div>
    <div>
      <button type="submit" name="type" value="fileupload">Datei hochladen</button>
    </div>
    <input type="hidden" name="formname" value=<?php
      echo $_POST[,formname'];
      ?>">
  </form>
  <form method="post">
    <button type="hidden" name="formname" value=<?php
      echo $_POST[,formname'];
      ?>">Zurück</button>
  </form>
</div>

```

```

text.php
<?php
$designation = ,';
$required = ,';
$plz = ,';
$max = ,';
if ( isset( $_POST[,id'] ) )
{
    foreach ($form as $output)
    {
        |
        if ( $_POST[,id'] == $output[,id'] )
        {
            $designation = $output[,designation'];
            $required = $output[,required'];
            $plz = $output[,plz'];
            $max = $output[,max'];
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != , )
{
    $designation = $_POST[,designation'];
}
if ( isset( $_POST[,required'] ) && $_POST[,required'] != , )
{
    $required = $_POST[,required'];
}
if ( isset( $_POST[,plz'] ) && $_POST[,plz'] != , )
{
    $plz = $_POST[,plz'];
}
if ( isset( $_POST[,max'] ) && $_POST[,max'] != , )
{
    $max = $_POST[,max'];
}
?>
<div class="linie">
    <form method="post">
        <div class="text">
            <label for="text">Bezeichnung:</label>
            <input type="text" name="designation" value="<?php
                echo $designation;
            ?>" required="required">
        </div>
        <div class="Pflichtfeld">
            <label>Pflichtfeld</label>
            <input type="checkbox" name="required" value="required"><?php
                if ( $required === ,required' ){
                    ?> checked="checked" <?php
                } ?> />
            </div>
        <div class="plz">
            <label>PLZ</label>
            <input type="checkbox" name="plz"><?php
                if ( $plz !== ,'){
                    ?> checked="checked" <?php
            
```

```

} ?> value="plz" />
</div>
<div class="Zeichen">
  <label>Anzahl Zeichen</label>
  <input type="number" name="max" value=<?php
echo $max;
?>" />
</div>
<input type="hidden" name="type" value="text">
<input type="hidden" name="formname" value=<?php
echo $_POST[,formname'];
?>">
<?php
if ( isset( $_POST[,id']) ){
  ?>
    <input type="hidden" name="id" value=<?php
echo $_POST[,id'];
?>"><?php
}
?>
  <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
  <button type="submit" name="formname" value=<?php
echo $_POST[,formname'];
?>">Zurück</button>
</form>
</div>

```

```

textarea.php
<?php
$designation = ,';
$required = ,';
$max = ,';
if ( isset( $_POST[,id'] ) )
{
    foreach ( $form as $output )
    {
        if ( $_POST[,id'] = $output[,id'] )
        {
            $designation = $output[,designation'];
            if ( isset( $output[,required'] ) )
            {
                $required = $output[,required'];
            }
            if ( isset( $output[,max'] ) )
            {
                $max = $output[,max'];
            }
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != ,')
{
    $designation = $_POST[,designation'];
}
if ( isset( $_POST[,required'] ) && $_POST[,required'] != ,')
{
    $required = $_POST[,required'];
}
if ( isset( $_POST[,max'] ) && $_POST[,max'] != ,')
{
    $max = $_POST[,max'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation" value="<?php
echo $designation;
?>" required="required">
</div>
<div class="Pflichtfeld">
    <label>Pflichtfeld</label>
    <input type="checkbox" name="required" value="required"<?php
if ( $required === ,required' )
{
    ?> checked="checked" <?php
} ?>>
</div>
<div class="Zeichen">
    <label>Max Zeichen</label>
    <input type="number" name="max" value="<?php
echo $max;
?>"/>

```

```
</div>
<input type="hidden" name="type" value="textarea">
<input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
<?php
if ( isset( $_POST[,id']) ){
?
?>
<input type="hidden" name="id" value="<?php
echo $_POST[,id'];
?>"><?php
}
?>
<button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
<button type="submit" name="formname" value="<?php
echo $_POST[,formname'];
?>">Zurück</button>
</form>
</div>
```

```

email.php
<?php
$designation = ,';
$required = ,';
$email = ,';
if ( isset( $_POST[,id'] ) )
{
    foreach ($form as $output)
    {
        if ( $_POST[,id'] == $output[,id'] )
        {
            $designation = $output[,designation'];
            $required = $output[,required'];
            if ( isset( $_POST[,email'] ) ){
                $email = $output[,email'];
            }
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != ,')
{
    $designation = $_POST[,designation'];
}
if ( isset( $_POST[,required'] ) && $_POST[,required'] != ,')
{
    $required = $_POST[,required'];
}
if ( isset( $_POST[,email'] ) && $_POST[,email'] != ,')
{
    $email = $_POST[,email'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation" value=<?php
        echo $designation;
    ?>" required="required">
</div>
<div class="Pflichtfeld">
    <label>Pflichtfeld</label>
    <input type="checkbox" name="required" value="required"<?php
        if ( $required === ,required' )
        {
            ?> checked="checked" <?php
        } ?>>
    </div>
    <input type="hidden" name="type" value="email">
    <input type="hidden" name="formname" value=<?php
        echo $_POST[,formname'];
    ?>"><?php
        if ( isset( $_POST[,id'] ) )
        {
            ?>
            <input type="hidden" name="id" value=<?php
                echo $_POST[,id'];
            ?>
        }
    </div>

```

```
?>><?php  
}  
?>  
<button type="submit" name="submit" value="save">Speichern</button>  
</form>  
<form method="post">  
    <button type="submit" name="formname" value="<?php  
        echo $_POST[.formname'];  
    ?>">Zurück</button>  
</form>  
</div>
```

```

date.php
<?php
$designation = ,';
$required = ,';
$min = ,';
$max = ,';
if ( isset( $_POST[,id'] ) )
{
    foreach ($form as $output)
    {
        if ( $_POST[,id'] == $output[,id'] )
        {
            $designation = $output[,designation'];
            $required = $output[,required'];
            $min = $output[,min'];
            $max = $output[,max'];
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != ,')
{
    $designation = $_POST[,designation'];
}
if ( isset( $_POST[,required'] ) && $_POST[,required'] != ,')
{
    $required = $_POST[,required'];
}
if ( isset( $_POST[,min'] ) && $_POST[,min'] != ,')
{
    $min = $_POST[,min'];
}
if ( isset( $_POST[,max'] ) && $_POST[,max'] != ,')
{
    $max = $_POST[,max'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation" value="<?php
echo $designation;
?>" required="required">
</div>
<div class="Pflichtfeld">
    <label>Pflichtfeld</label>
    <input type="checkbox" name="required" value="required"<?php
if ( $required === ,required' )
{
    ?> checked="checked" <?php
} ?>>
</div>
<div class="text">
    <label>Datum von:</label>
    <input type="date" name="min" value="<?php
echo $min;
?>" pattern="(0[1-9]|1[0-9]|2[0-9]|3[01]).(0[1-9]|1[012]).[0-9]{4}" placeholder="tt.mm.jjjj">

```

```

</div>
<div class="text">
    <label>Datum bis:</label>
    <input type="date" name="max" value="<?php
        echo $max;
    ?>" pattern="(0[1-9]|1[0-9]|2[0-9]|3[01]).(0[1-9]|1[012]).[0-9]{4}" placeholder="tt.mm.jjjj">
</div>
    <input type="hidden" name="type" value="date">
    <input type="hidden" name="formname" value="<?php
        echo $_POST[,formname'];
    ?>">
    <?php
    if ( isset( $_POST[,id']) )
    {
        ?>
        <input type="hidden" name="id" value="<?php
            echo $_POST[,id'];
        ?>"><?php
    }
    ?>
    <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
    <button type="submit" name="formname" value="<?php
        echo $_POST[,formname'];
    ?>">Zurück</button>
</form>
</div>

```

```

number.php
<?php
$designation = ,';
|required = ,';
$min = ,';
$max = ,';
if ( isset( $_POST[,id'] ) )
{
    foreach ($form as $output)
    {
        if ( $_POST[,id'] == $output[,id'] )
        {
            $designation = $output[,designation'];
            $required = $output[,required'];
            $min = $output[,min'];
            $max = $output[,max'];
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != ,')
{
    $designation = $_POST[,designation'];
}
if ( isset( $_POST[,required'] ) && $_POST[,required'] != ,')
{
    $required = $_POST[,required'];
}
if ( isset( $_POST[,min'] ) && $_POST[,min'] != ,')
{
    $min = $_POST[,min'];
}
if ( isset( $_POST[,max'] ) && $_POST[,max'] != ,')
{
    $max = $_POST[,max'];
}
?>
<div class="linie">
    <form method="post">
        <div class="text">
            <label>Bezeichnung:</label>
            <input type="text" name="designation" value="<?php
                echo $designation;
            ?>" required="required">
        </div>
        <div class="Pflichtfeld">
            <label>Pflichtfeld</label>
            <input type="checkbox" name="required" value="required"<?php
                if ( $required === ,required' )
                {
                    ?> checked="checked" <?php
                }
            ?>>
        </div>
        <div class="Min">
            <label>Min. wert</label>
            <input type="number" name="min" value="<?php
                echo $min;
            ?>">
        </div>
    </form>
</div>

```

```

?>“>
</div>
<div class=“Max“>
    <label>Max. wert</label>
    <input type=“number“ name=“max“ value=<?php
        echo $max;
    ?>“>
</div>
<input type=“hidden“ name=“type“ value=“number“>
<input type=“hidden“ name=“formname“ value=<?php
        echo $_POST[,formname'];
    ?>“><?php
if ( isset( $_POST[,id']) ) )
{
    ?>
    <input type=“hidden“ name=“id“ value=<?php
        echo $_POST[,id'];
    ?>“><?php
}
?>
<button type=“submit“ name=“submit“ value=“save“>Speichern</button>
</form>
<form method=“post“>
    <button type=“submit“ name=“formname“ value=<?php
        echo $_POST[,formname'];
    ?>“>Zurück</button>
</form>
</div>

```

```

fileupload.php
<?php
$designation = ,';
$required = ,';
if ( isset( $_POST[,id']) )
{
    foreach ($form as $output)
    {
        if ( $_POST[,id'] == $output[,id'] )
        {
            $designation = $output[,designation'];
            $required = $output[,required'];
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != ,')
{
    $designation = $_POST[,designation'];
}
if ( isset( $_POST[,required'] ) && $_POST[,required'] != ,')
{
    $required = $_POST[,required'];
}
?>
<div class="linie">
    <form method="post">
        <div class="text">
            <label>Bezeichnung:</label>
            <input type="text" name="designation" value="<?php
                echo $designation;
            ?>" required="required">
        </div>
        <div class="Pflichtfeld">
            <label>Pflichtfeld</label>
            <input type="checkbox" name="required" value="required"<?php
                if ( $required === ,required')
                {
                    ?> checked="checked" <?php
                }
            ?>>
        </div>
        <input type="hidden" name="type" value="fileupload">
        <input type="hidden" name="formname" value="<?php
            echo $_POST[,formname'];
        ?>">
        <?php
        if ( isset( $_POST[,id']) )
        {
            ?>
            <input type="hidden" name="id" value="<?php
                echo $_POST[,id'];
            ?>"><?php
        }
        ?>
        <button type="submit" name="submit" value="save">Speichern</button>
    </form>

```

```
<form method="post">
  <button type="submit" name="formname" value="<?php
    echo $_POST[,formname'];
    ?>">Zurück</button>
</form>
</div>
```

```

radiobox.php
<?php
$designation = ,;
if ( isset( $_POST[,id']) )
{
    foreach ($form as $output) {
        if ($output[,type'] === ,radiobox')
        {
            if ( isset($_POST[,fk_id']) && $_POST[,fk_id'] = $output[,id'] )
            {
                $designation = $output[,designation'];
            }
            elseif ( $_POST[,id'] = $output[,id'] )
            {
                $designation = $output[,designation'];
            }
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != , )
{
    $designation = $_POST[,designation'];
}
?>
<div class="linie">
    <form method="post">
        <div class="text">
            <label>Bezeichnung:</label>
            <input type="text" name="designation" value="<?php
echo $designation;
?>" required="required">
        </div>
        <input type="hidden" name="type" value="radiobox">
        <input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
        <?php
if ( isset( $_POST[,id']) )
{
    ?>
    <input type="hidden" name="id" value="<?php
if ( isset( $newid) )
{
    echo $newid;
}
else
{
    echo $_POST[,id'];
}
?>"><?php
}
?>
        <button type="submit" name="submit" value="save">Speichern</button>
    </form>
    <form method="post">
        <button type="submit" name="formname" value="<?php echo $_POST[,formname']; ?>">Zurück</button>
    </form>

```

```

<?php
if ( isset( $newid ) || isset( $_POST[,id'] ) )
{
?>
<form method="post">
    <button NAME="plus" type="submit">
        Radiobox Erstellen
    </button>
    <input type="hidden" name="fk_id" value="<?php
if ( isset( $_POST[,fk_id'] ) && $_POST[,fk_id'] !== ',' )
{
    echo $_POST[,fk_id'];
}
elseif ( isset( $newid ) && $newid != ',' )
{
    echo $newid;
}
else{
    echo $_POST[,id'];
}
?>">
    <input type="hidden" name="type" value="radio">
    <input type="hidden" name="formname" value="<?php echo $_POST[,formname']; ?>">
</form>
<?php
foreach ($form as $output)
{
    if ( isset($output[,fk_id']) && isset( $_POST[,id'] ) && $_POST[,id'] === $output[,fk_id'] )
    {
?>
        <form method="post">
            <button type="submit" name="submit" value="edit"><?php
echo $output[,designation'];
?><!-<input type="radio"/>-> Radiobox</button>
            <div class="text">
                <button type="submit" name="submit" value="delete">Löschen</button>
                <input type="hidden" name="id" value="<?php
echo $_POST[,id'];
?>">
                <input type="hidden" name="fk_id" value="<?php
echo $_POST[,fk_id'];
?>">
            </div>
            <input type="hidden" name="type" value="check">
            <input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>"><?php
if ( isset( $_POST[,id'] ) )
{
?>
            <input type="hidden" name="id" value="<?php
echo $output[,id'];
?>">
            <input type="hidden" name="fk_id" value="<?php
echo $_POST[,fk_id'];
?>"><?php
}
?>
        </form>
    <?php
}
?></div>

```

```

radio.php
<?php
$designation = ,';
if ( isset( $_POST[,id']) )
{
    foreach ($form as $output)
    {
        if ( $_POST[,id'] == $output[,id'] )
        {
            $designation = $output[,designation'];
        }
    }
}
if ( isset( $_POST[,designation2'] ) && $_POST[,designation2'] != ,')
{
    $designation = $_POST[,designation2'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation2" value="<?php
echo $designation;
?>" required="required">
</div>
<input type="hidden" name="type" value="radio">
<input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
<input type="hidden" name="fk_id" value="<?php
echo $_POST[,fk_id'];
?>"><?php
if ( isset( $_POST[,id']) )
{
    ?>
    ?>
    <input type="hidden" name="id" value="<?php
echo $_POST[,id'];
?>"><?php
}
?>
    <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
    <button type="submit" name="formname" value="<?php
echo $_POST[,formname'];
?>">Zurück</button>
    <input type="hidden" name="id" value="<?php
echo $output[,fk_id'];
?>">
    <input type="hidden" name="type" value="radiobox">
</form>
</div>

```

```

checkbox.php
<?php
$designation = ,;
if ( isset( $_POST[,id'] ) )
{
    foreach ($form as $output)
    {
        if ($output[,type'] === ,checkbox')
        {
            if ( isset($_POST[,fk_id']) && $_POST[,fk_id'] = $output[,id'] )
            {
                $designation = $output[,designation'];
            }
            elseif ( $_POST[,id'] = $output[,id'] )
            {
                $designation = $output[,designation'];
            }
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != , )
{
    $designation = $_POST[,designation'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation" value="<?php
echo $designation;
?>" required="required">
</div>
    <input type="hidden" name="type" value="checkbox">
    <input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
<?php
if ( isset( $_POST[,id'] ) )
{
    ?>
    <input type="hidden" name="id" value="<?php
if ( isset( $newid) )
{
    echo $newid;
}
else
{
    echo $_POST[,id'];
}
?>"><?php
}
?>
    <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
    <button type="submit" name="formname" value="<?php echo $_POST[,formname']; ?>">Zurück</button>

```

```

</form>
<?php
if ( isset( $newid ) || isset( $_POST[,id'] ) )
{
?
<form method="post">
<button NAME="plus" type="submit">
    Checkbox Felder Erstellen
</button>
<input type="hidden" name="fk_id" value="<?php
if ( isset( $_POST[,fk_id'] ) && $_POST[,fk_id'] !== ',' )
{
    echo $_POST[,fk_id'];
}
elseif ( isset( $newid ) && $newid != ',' )
{
    echo $newid;
}
else
{
    echo $_POST[,id'];
}
?>">
<input type="hidden" name="type" value="check">
<input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
</form><?php
foreach ($form as $output)
{
if ( isset($output[,fk_id']) ) && isset( $_POST[,id'] ) && $_POST[,id'] === $output[,fk_id'] )
{
?
<form method="post">
<button type="submit" name="submit" value="edit"><?php
echo $output[,designation'];
?> Checkbox</button>
<div class="text">
    <button type="submit" name="submit" value="delete">Löschen</button>
    <input type="hidden" name="id" value="<?php
echo $_POST[,id'];
?>">
    <input type="hidden" name="fk_id" value="<?php
echo $_POST[,fk_id'];
?>">
</div>
<input type="hidden" name="type" value="check">
<input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>"><?php
if ( isset( $_POST[,id'] ) )
{
?
<input type="hidden" name="id" value="<?php
echo $output[,id'];
?>">
    <input type="hidden" name="fk_id" value="<?php
echo $_POST[,fk_id'];
?>"><?php
}
?
</form>
<?php
}

```

```

        }
    }
?>
</div>

check.php
<?php
$designation = ,';
if ( isset( $_POST[,id']) )
{
    foreach ( $form as $output )
    {
        if ( $_POST[,id'] = $output[,id'] )
        {
            $designation = $output[,designation'];
        }
    }
}
if ( isset( $_POST[,designation2'] ) && $_POST[,designation2'] != ,')
{
    $designation = $_POST[,designation2'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label><input type="text" name="designation2" value="<?php
echo $designation;
?>" required="required">
</div>
    <input type="hidden" name="type" value="check">
    <input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>"><?php
if ( isset( $_POST[,id']) )
{
    ?>
    <input type="hidden" name="id" value="<?php
echo $_POST[,id'];
?>"><?php
}
?>
    <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
    <button type="submit" name="formname" value="<?php
echo $_POST[,formname'];
?>">Zurück</button>
    <input type="hidden" name="id" value="<?php
echo $output[,fk_id'];
?>">
    <input type="hidden" name="type" value="radiobox">
</form>
</div>

```

```

select.php
<?php
$designation = ,;
if ( isset( $_POST[,id']) )
{
    foreach ($form as $output)
    {
        if ($output[,type'] === ,select')
        {
            if ( isset($_POST[,fk_id']) && $_POST[,fk_id'] == $output[,id'] )
            {
                $designation = $output[,designation'];
            }
            elseif ( $_POST[,id'] == $output[,id'] )
            {
                $designation = $output[,designation'];
            }
        }
    }
}
if ( isset( $_POST[,designation'] ) && $_POST[,designation'] != , ){
    $designation = $_POST[,designation'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation" value="<?php
echo $designation;
?>" required="required">
</div>
    <input type="hidden" name="type" value="select">
    <input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
    <?php
if ( isset( $_POST[,id']) )
{
    ?>
    <input type="hidden" name="id" value="<?php
if ( isset( $newid) )
{
    echo $newid;
}
else
{
    echo $_POST[,id'];
}
?>"><?php
}
?>
    <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
    <button type="submit" name="formname" value="<?php
echo $_POST[,formname'];"
```

```

?>“>Zurück</button>
</form>
<?php
if ( isset( $newid ) || isset( $_POST[,id'] ) )
{
?
<form method="post">
    <button NAME="plus" type="submit">
        Auswahl Erstellen
    </button>
    <input type="hidden" name="fk_id" value="<?php
if ( isset( $_POST[,fk_id'] ) && $_POST[,fk_id'] !== ',' )
{
    echo $_POST[,fk_id'];
}
elseif ( isset( $newid ) && $newid != ',' )
{
    echo $newid;
}
else
{
    echo $_POST[,id'];
}
?>">
    <input type="hidden" name="type" value="option">
    <input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
</form><?php
foreach ($form as $output)
{
if ( isset($output[,fk_id']) && isset( $_POST[,id'] ) && $_POST[,id'] === $output[,fk_id'] )
{
?
    <form method="post">
        <button type="submit" name="submit" value="edit"><?php
        echo $output[,designation'];
        ?> Auswahlliste</button>
        <div class="text">
            <button type="submit" name="submit" value="delete">Löschen</button>
            <input type="hidden" name="id" value="<?php
            echo $_POST[,id']; ?>">
        </div>
        <input type="hidden" name="fk_id" value="<?php
        echo $_POST[,fk_id']; ?>">
        <input type="hidden" name="type" value="check">
        <input type="hidden" name="formname" value="<?php
        echo $_POST[,formname'];
        ?>"><?php
if ( isset( $_POST[,id'] ) )
{
?
    <input type="hidden" name="id" value="<?php
    echo $output[,id'];
    ?>">
    <input type="hidden" name="fk_id" value="<?php
    echo $_POST[,fk_id'];
    ?>"><?php
}
?
    </form><?php
}
}
?></div>

```

```

option.php
<?php
$designation = ,';
if ( isset( $_POST[,id'] ) )
{
    foreach ($form as $output)
    {
        if ( $_POST[,id'] = $output[,id'] )
        {
            $designation = $output[,designation'];
        }
    }
}
if ( isset( $_POST[,designation2'] ) && $_POST[,designation2'] != , )
{
    $designation = $_POST[,designation2'];
}
?>
<div class="linie">
<form method="post">
<div class="text">
    <label>Bezeichnung:</label>
    <input type="text" name="designation2" value="<?php
echo $designation;
?>" required="required">
</div>
<input type="hidden" name="type" value="option">
<input type="hidden" name="formname" value="<?php
echo $_POST[,formname'];
?>">
<input type="hidden" name="fk_id" value="<?php
echo $_POST[,fk_id'];
?>"><?php
if ( isset( $_POST[,id'] ) )
{
?>
<input type="hidden" name="id" value="<?php
echo $_POST[,id'];
?>"><?php
?>
    <button type="submit" name="submit" value="save">Speichern</button>
</form>
<form method="post">
    <button type="submit" name="formname" value="<?php
echo $_POST[,formname'];
?>">Zurück</button>
    <input type="hidden" name="id" value="<?php
echo $output[,fk_id'];
?>">
    <input type="hidden" name="type" value="select">
</form>
</div>

```

### **output.php**

```
<?php
include('function.php');
include '../form/'. $_GET[formname].'.php';
$form = unserialize($form);
$send = 0;
if ( isset( $_POST[send] ) )
{
    $message = '';
    foreach ($form as $output)
    {
        switch ($output[type])
        {
            case ,number':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. $_POST[ $output[id] ].'<br />';
                }
                break;
            case ,email':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. $_POST[ $output[id] ].'<br />';
                }
                break;
            case ,textarea':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. $_POST[ $output[id] ].'<br />';
                }
                break;
            case ,text':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. $_POST[ $output[id] ].'<br />';
                }
                break;
            case ,date':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. $_POST[ $output[id] ].'<br />';
                }
                break;
            case ,radiobox':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. $_POST[ $output[id] ].'<br />';
                }
                break;
            case ,checkbox':
                if ( isset( $_POST[ $output[id] ] ) )
                {
                    $message .= $output[designation].':'. '<br />';
                    foreach ( $_POST[ $output[id] ] as $item){
                        $message .= $item.'<br />';
                    }
                }
        }
    }
}
```

```

        }
        break;
    case ,select‘:
        if ( isset( $_POST[ $output[,id‘] ] ) )
        {
            $message .= $output[,designation‘].‘:‘<br />‘;
            foreach ( $_POST[ $output[,id‘] ] as $item){
                $message .= $item.‘<br />‘;
            }
        }
        break;
    }
}

include ,sendmail.php‘;
$send = 0;
}
if ( $send == 1 )
{
?>
Formular wurde versendet.
<?php
}
else
{
    if ($send == 2)
    {
        ?>
        Das Dokument ist zu gross um es hochzuladen.
        <?php
    }
?>
<html>
<head>
    <link href= „/css/design/formularstyle.css“ rel=“stylesheet“>
    <title>Ausgabe</title>
</head>
<body>
<div class=“wrapper“>
    <form method=“post“ enctype=“multipart/form-data“><?php
        foreach ($form as $output)
        {
            switch ($output[,type‘]) {
                case ,number‘:
                    ?>
                    <div class=“number“>
                        <label><?php
                            echo $output[,designation‘];
                        ?></label>
                        <input type=“number“ name=“<?php
                            echo $output[,id‘];
                        ?>“ id=“text“<?php
                            if ( $output[,max‘] != ,‘ )
                            {
                                ?>maxlength=“<?php
                                    echo $output[,max‘];
                                ?>“<?php
                            }
                            if ( $output[,required‘] === ,required‘ )
                            {
                                ?> required=“required“ <?php
                            } ?> value=“<?php
                            if ( isset( $_POST[ $output[,id‘] ] ) )
                            {

```

```

        echo $_POST[ $output[,id' ] ];
    }
?>“>
</div>
<?php
break;
case ,email‘:
<div class=“email“>
<label><?php
echo $output[,designation‘];
?></label>
<input type=“email“ name=“<?php
echo $output[,id‘];
?>“ <?php
if ( $output[,required‘] === ,required‘ )
{
    ?> required=“required“ <?php
} ?> value=“<?php
if ( isset( $_POST[ $output[,id‘] ] ) )
{
    echo $_POST[ $output[,id‘] ];
}
?>“ pattern=“[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$“>
</div>
<?php
break;
case ,textarea‘:
?>
<div class=“textarea“>
<label><?php
echo $output[,designation‘];
?></label>
<textarea cols=“23“ rows=“5“ name=“<?php
echo $output[,id‘];
?>“ <?php
if ( $output[,required‘] === ,required‘ )
{
    ?> required <?php
} ?> value=“<?php
if ( isset( $_POST[ $output[,id‘] ] ) )
{
    echo $_POST[ $output[,id‘] ];
}
?>“><?php
if ( $output[,max‘] != ,‘ )
{
    ?>maxlength=“<?php
echo $output[,max‘];
?>“<?php
}
?></textarea>
</div>
<?php
break;
case ,text‘:
?>
<div class=“text“>
<label><?php
echo $output[,designation‘];
?></label>
<input type=“text“ name=“<?php
echo $output[,id‘];
?>“ <?php
if ( $output[,plz‘] != ,‘ )

```

```

{
    ?> pattern="\d*“<?php
}
if ( $output[,max'] != ,‘ )
{
    ?>maxlength="“<?php
    echo $output[,max'];
    ?>“<?php
}
if ( $output[,required'] === ,required' )
{
    ?> required="required“ <?php
} ?> value="“<?php
if ( isset( $_POST[ $output[,id'] ] ) )
{
    echo $_POST[ $output[,id'] ];
}
?>“>
</div>
<?php
break;
case ,date':
?>
<div class="date">
<label><?php
echo $output[,designation'];
?></label>
<input type="date" name="“<?php
echo $output[,id'];
?>“ <?php
if ( $output[,required'] === ,required' )
{
    ?>required="required“ <?php
} ?>value="“<?php
if ( isset( $_POST[ $output[,id'] ] ) )
{
    echo $_POST[ $output[,id'] ];
}
?>“ <?php
if ( $output[,min'] != ,‘ )
{
    ?>min="“<?php
    echo $output[,min'];
    ?>“<?php
}
if ( $output[,max'] != ,‘ )
{
    ?>min="“<?php
    echo $output[,max'];
    ?>“<?php
}
?> pattern="(0[1-9]|1[0-9]|2[0-9]|3[01]).(0[1-9]|1[012]).[0-9]{4}“ placeholder="tt.mm.jjjj“>
</div>
<?php
break;
case ,fileupload':
?>
<div class="fileupload">
<label><?php
echo $output[,designation'];
?></label>
<input type="file" name="file[]">
</div>
<?php

```

```

        break;
    case ,radiobox':
        $id = $output[,'id'];
    ?>
    <div class="radiobox">
        <fieldset>
            <legend><?php
                echo $output[,'designation'];
            ?></legend><?php
                foreach ($form as $output)
            {
                if ( isset($output[,'fk_id']) && $output[,'fk_id'] == $id)
                {
                    ?>
                    <input type="radio" name=<?php
                        echo $id; ?>" value=<?php
                        echo $output[,'designation'];
                    ?>" <?php
                        if ( isset($output[,'required']) && $output[,'required'] === ,required' )
                        {
                            ?>required="required" <?php
                        } ?>/>
                    <label class="nofloat"><?php
                        echo $output[,'designation'];
                    ?></label>
                    <?php
                }
            }
            ?>
        </fieldset>
    </div>
<?php
break;
case ,checkbox':
    ?>
    <div class="checkbox">
        <fieldset>
            <legend><?php
                echo $output[,'designation'];
            ?></legend>
        </fieldset><?php
        $id = $output[,'id'];
        foreach ($form as $output)
        {
            if ( isset($output[,'fk_id']) && $output[,'fk_id'] == $id)
            {
                ?>
                <input type="checkbox" name=<?php echo $id; ?>[] value=<?php echo $output[,'designation'];
            }
        }
        ?>
        <label class="nofloat"><?php
            echo $output[,'designation'];
        ?></label><br/><?php
    }
}
?>
</div>
<?php
break;
case ,select':
    $id = $output[,'id'];
    ?>
    <div class="select">
        <label for="select">select</label>
        <select name=<?php

```

```

echo $output['id'];
?>[]“ multiple=“multiple“><?php
foreach ($form as $output)
{
    if ( isset($output['fk_id']) && $output['fk_id'] == $id)
    {
        ?>
        <option value=<?php
        echo $output['designation'];
?>“><?php
        echo $output['designation'];
?></option><?php
    }
}
?>
</select>
</div><?php
break;
}
}
?>
<button type=“submit“ name=“send“>Abschicken</button>
</form>
</div>
</body>
</html>
<?php
}
?>

```

```

sendmail.php
<?php
$recipient_email = $form[0][,email']; //recipient
$from_email = „info@your_domain.com“; //from email using site domain.
$subject = $form[0][,subject'];//email subject line
$sender_name = filter_var(,Formular', FILTER_SANITIZE_STRING); //capture sender name
$sender_email = filter_var($form[0][,reply'], FILTER_SANITIZE_STRING); //capture sender email
$sender_message = filter_var($message, FILTER_SANITIZE_STRING); //capture message

if ( isset($_FILES[,file']) ){
    $attachments = $_FILES[,file'];
    $file_count = count($attachments[,name']); //count total files attached
}
//php validation
if (strlen($sender_name)<4){
die(,Name is too short or empty');
}
if (!filter_var($sender_email, FILTER_VALIDATE_EMAIL)) {
die(,Invalid email');
}
$boundary = md5(„sanwebe.com“);
if ($file_count > 0){ //if attachment exists
    //header
    $headers = „MIME-Version: 1.0\r\n“;
    $headers .= „From:“.$from_email.“\r\n“;
    $headers .= „Reply-To: ..$sender_email.“ ..\r\n“;
    $headers .= „Content-Type: multipart/mixed; boundary = $boundary\r\n\r\n“;
    //message text
    $body = „--$boundary\r\n“;
    $body .= „Content-Type: text/plain; charset=ISO-8859-1\r\n“;
    $body .= „Content-Transfer-Encoding: base64\r\n\r\n“;
    $body .= chunk_split(base64_encode($sender_message));
    if ( isset($_FILES[,file']) ){
        //attachments
        for ($x = 0; $x < $file_count; $x++){
            if (!empty($attachments[,name'][$x])){
                if ($attachments[,error'][$x]>0) //exit script and output error if we encounter any
                {
                    $mymsg = array(
                        1=>“The uploaded file exceeds the upload_max_filesize directive in php.ini“,
                        2=>“The uploaded file exceeds the MAX_FILE_SIZE directive that was specified in the HTML form“,
                        3=>“The uploaded file was only partially uploaded“,
                        4=>“No file was uploaded“,
                        6=>“Missing a temporary folder“ );
                    die($mymsg[$attachments[,error'][$x]]);
                }
                //get file info
                $file_name = $attachments[,name'][$x];
                $file_size = $attachments[,size'][$x]/1000000;
                if ( isset( $form[0][,max'] ) && $form[0][,max'] < $file_size ){
                    $send = 2;
                }
                $file_type = $attachments[,type'][$x];
                //read file
                $handle = fopen($attachments[,tmp_name'][$x], „r“);
                $content = fread($handle, $file_size);
                fclose($handle);
            }
        }
    }
}

```

```

$encoded_content = chunk_split(base64_encode($content)); //split into smaller chunks (RFC 2045)
$body .= "-$boundary\r\n";
$body .= "Content-Type: $file_type; name=\"$file_name\"\r\n";
$body .= "Content-Disposition: attachment; filename=\"$file_name\"\r\n";
$body .= "Content-Transfer-Encoding: base64\r\n";
$body .= "X-Attachment-Id: ..rand(1000,99999).\"r\n\r\n";
$body .= $encoded_content;
}
}
}
}
else
{//send plain email otherwise
if ($send == 0)
{
$headers = "From:".$from_email."\r\n".
"Reply-To: ".$sender_email. "\n".
"X-Mailer: PHP/" . phpversion();
$body = $sender_message;
}
}
if ($send == 0)
{
$sentMail = @mail($recipient_email, $subject, $body, $headers);
}
}
?>

```

```

style.css
html, body, form, fieldset, legend {
    margin: 0;
    padding: 0;
}

body {
    background-color: #0099a8;
}

.wrapper{
    width: 820px;
    font-family: Arial, „Times New Roman“, Times, serif;
    font-size: 13px;
    color: black;
    margin: 10px auto;
}

.main{
    padding:60px;
    background-color: white;
    min-height: 200px;
}

.liniel{
    border-radius: 5px;
    line-height: 30px;
    padding: 5px 10px;
    border: 2px solid #0099a8;
    margin-bottom: 5px;
}

form {
    width: 675px;
    background: #FFFFFF;
    border-radius: 5px;
    padding: 5px;
    margin: auto;
}

h2 {
    display: table;
    margin-left: auto;
    margin-right: auto;
}

img {
    float:right;
    margin: 10px 10px 30px 0px;
}

fieldset {
    border: none;
    margin-bottom: 10px;
}

fieldset { margin: 0; }

```

```

form div {
    border-radius: 5px;
    line-height: 30px;
    padding: 5px 10px;
    margin-bottom: 5px;
}

label {
    float: left;
    width: 210px;
    text-align: right;
    cursor: pointer;
    margin-right: 10px;
}

label.nofloat {
    float: none;
    width: auto;
    text-align: left;
}

input, textarea, select {
    color: #333;
    font-style: italic;
    font-size: 13px;
    font-family: Georgia, „Times New Roman“, Times, serif;
    background: white;
    border: 2px solid #0099a8;
    border-radius: 3px;
    padding: 5px;
    outline: none;
    width: 200px;
}

input:focus, textarea:focus { background: #eaeaea; }

.radiobox input { width: auto; }
.select { margin-bottom: 3px; }
.checkbox input {
    width: auto;
    margin-left: 120px;
}
.checkbox a { color: black; }

button {
    display: block;
    width: 300px;
    background: #d1b400;
    color: #ffffff;
    font-size: 14px;
    letter-spacing: 1px;
    padding: 10px 15px;
    border: 2px solid #0099a8;
    border-radius: 20px;
    margin: auto;
    font-weight: bolder;
}

button:hover {
    background: #FFFFFF;
    color: #000000;
    cursor: pointer;
}

```

```

formularstyle.css
html, body, form, fieldset, legend {
    margin: 0;
    padding: 0;
}

body {
    background-color: #FFFFFF;
}

.wrapper{
    width: 820px;
    font-family: Arial, „Times New Roman“, Times, serif;
    font-size: 13px;
    background-color: white;
    color: black;
    margin: 10px auto;
}

form {
    width: 660px;
    background: #FFFFFF;
    border-radius: 5px;
    padding: 5px;
    margin: auto;
}

fieldset {
    border: none;
    margin-bottom: 10px;
}

fieldset { margin: 0; }

form div {
    border-radius: 5px;
    line-height: 30px;
    padding: 5px 10px;
    margin-bottom: 5px;
}

label {
    float: left;
    width: 210px;
    text-align: right;
    cursor: pointer;
    margin-right: 10px;
}

label.nofloat {
    float: none;
    width: auto;
    text-align: left;
}

```

```

fieldset legend {
    float: left;
    width: 210px;
    text-align: right;
    font-size: 13px;
    font-weight: normal;
    padding-bottom: 0;
    margin-top: 0;
    margin-right: 10px;
}

input, textarea, select {
    color: #333;
    font-style: italic;
    font-size: 13px;
    font-family: Georgia, „Times New Roman“, Times, serif;
    background: white;
    border: 2px solid #000000;
    border-radius: 3px;
    padding: 5px;
    outline: none;
    width: 200px;
}

input:focus, textarea:focus { background: #eaeaea; }

.radiobox input { width: auto; }
.select { margin-bottom: 3px; }
.checkbox input {
    width: auto;
    margin-left: 220px;
}
.checkbox a { color: black; }

button {
    display: block;
    width: 300px;
    background: #d1b400;
    color: #ffffff;
    font-size: 14px;
    letter-spacing: 1px;
    padding: 10px 15px;
    border: 2px solid #0099a8;
    border-radius: 20px;
    margin: auto;
}
button:hover {
    background: #FFFFFF;
    color: #000000;
    cursor: pointer;
}

```