

Landesberufsschule für Handwerk und Industrie Bozen



Maturaklasse mit Fachbereich Informatik

Maturaprojekt im Rahmen der Staatlichen Abschlussprüfung

Thema:

Spendenwebsite für Hilfsbedürftige



Schuljahr 2022/23

Autor der Projektarbeit:

Samuel Soini

Tutor:

Wilhelm Lanz

Abgabe:

06.06.2023



Inhaltsverzeichnis

1. Vorwort	3
2. Projektplanung	3
2.1 Nutzwertanalyse	3
2.2 Wettbewerbsanalyse	4
2.3 Soll-Zustand	5
3. Ausgangssituation	6
3.1 Projektziel	6
3.2 Projektumfeld	6
3.3 Ist-Zustand	7
3.4 Verwendete Technologien	7
4. Zeitplanung und Management	9
4.1 Projektentwicklungsmethode	9
4.2 Zeitplanung	9
5. Projektumsetzung	10
5.1 Design/Frontend Entwicklung	10
5.2 Backend Entwicklung	12
5.3 Datenbank Entwicklung	13
6. Projektergebnisse	14
6.1 Schwierigkeiten und Hindernisse	14
6.2 Ist-Soll Vergleich Funktionen und Zeitplanung	14
6.3 Fazit	15
7. Selbständigkeitserklärung	16
8. Quellen	16

1. Vorwort

In der Maturaklasse der LBSHI in Bozen ist es verpflichtend ein fachbezogenes Projekt im Laufe des Schuljahres zu entwickeln und zu guter Letzt vorzutragen. Somit haben Markus Mair und ich uns entschieden, zusammen ein Projekt zu entwickeln welches uns über das ganze Schuljahr auf Trapp hält und auf welchen wir in Zukunft stolz sein können. Ich möchte auch einen Dank an Markus aussprechen, der während der Entwicklung stets hinter unserem Projekt stand und genauso viel Herzblut hineingesteckt hat wie ich es tat.

2. Projektplanung

Die Grundidee von „dono“ ist es für Hilfsbedürftige zu spenden. Uns war es wichtig eine dazu passende Webseite zu kreieren. Doch um dies zu bewältigen war es noch notwendig das Projekt von Grund auf zu planen. Zum einen wurde eine Nutzwertanalyse gemacht, um zu sehen, wie gut unsere Idee im Vergleich zu anderen möglichen Projekten ist, zum anderen eine Wettbewerbsanalyse samt einen Soll-Zustand, um zu verdeutlichen, wie unser Projekt am Ende sein soll.

2.1 Nutzwertanalyse

Anfangs standen wir zwischen zwei möglichen Projekten, welche sich in ihrem Sinn und Zweck vollkommen unterscheiden. Zum einen das Erstellen einer Spendenseite welche letztendlich umgesetzt wurde, zum anderen eine Workout- bzw. Fitness- App, welche es ermöglicht Workout Pläne zu erstellen und individuell zu personalisieren.

Nutzwertanalyse		Workout App		Charity	
Kriterium	Gewicht	A	B	A	B
Ges. Nutzen	45%	30	13,5	60	22
Aufwand	25%	50	12,5	35	8,75
Spaß	10%	20	2	15	1,5
Lernfaktor	10%	50	5	40	4
Erlaubt Geld	10%	30	3	30	3
Ges.		36	44,25		

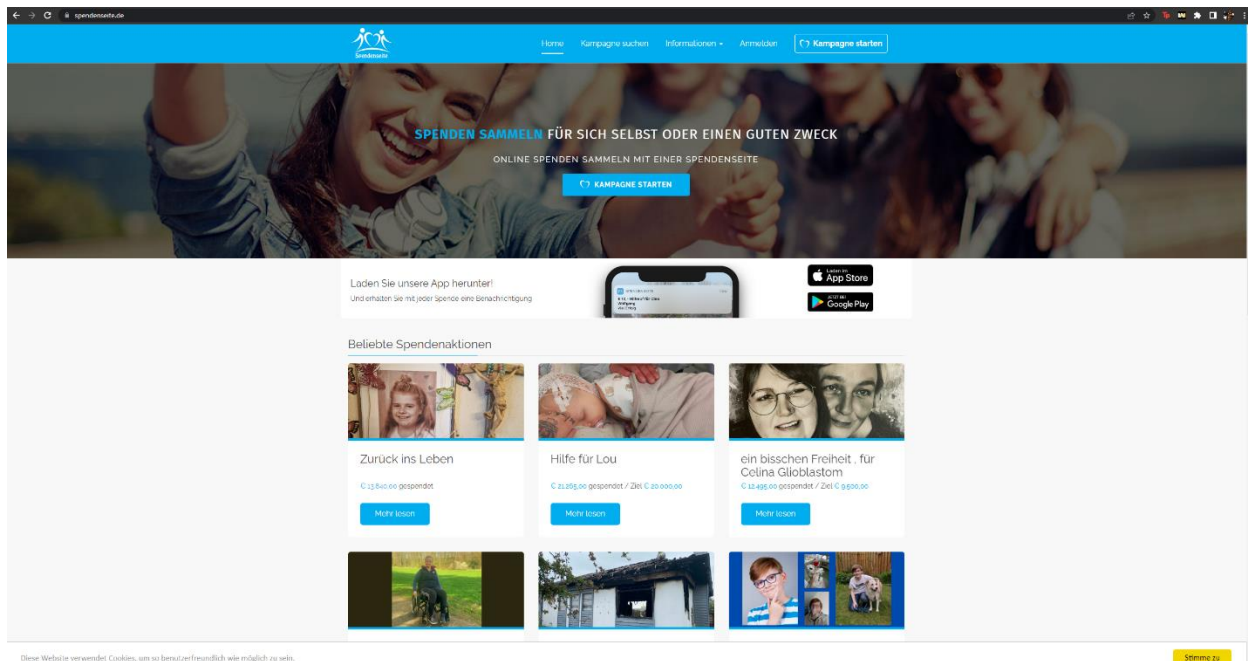
Bei dieser Nutzwertanalyse legten wir großen Wert auf:

- Nutzen für die Gesellschaft
- Aufwand
- Weiterbildung/Weiterentwicklung

Daraufhin fiel die Entscheidung letztendlich auf die Spendenseite „dono“ da sie einen deutlich größeren gesellschaftlichen Nutzen erbringt. Zudem wollten wir erlerntes Wissen vom letzten Jahr vertiefen, anstatt uns direkt in eine neuen Programmierumgebung zu stürzen.

2.2 Wettbewerbsanalyse

Da wir nun unser Projekt - anhand einer Nutzwertanalyse - ausgewählt haben, mussten wir uns nun auf die Wettbewerbsanalyse konzentrieren. Grundsätzlich gibt es in dieser Form nicht viele Websites da es sich bei den meisten Sites um Spendenaktionen für die eigene Organisation handelt. Dabei haben wir uns auf die Website spendenseite.de bezogen, die in Deutschland vertreten ist, da diese am vergleichbarsten zu unserer Website ist. Der Unterschied besteht darin, dass ihr Umsatz dadurch erzielt wird, dass sie zwischen dem Spender und dem Kampagnenbetreiber stehen und einen Teil der Spende erhalten, während wir lediglich unser Geld durch freiwilliges Spenden verdienen möchten.





Was ist eine Wettbewerbsanalyse?

Bei einer Wettbewerbsanalyse geht es darum Konkurrenten zu suchen und zu analysieren, um folgende Kernfragen zu beantworten:

- Was hat die Konkurrenz entwickelt?
- Was gefällt mir gut/was weniger?
- Welche Funktionen beinhaltet deren Projekt?
- Was kann ich anders machen?
- usw.

2.3 Soll-Zustand

Nachdem wir unser Projekt ausgewählt und eine Wettbewerbsanalyse durchgeführt haben, konzentrierten wir uns auf Funktionen, die eingebaut werden sollten.

Unsere geplanten Funktionen waren:

- Spenden über PayPal ← idealerweise PaypalMe
- Spenden über Direktüberweisung
- Anonyme Spenden
- Spendenziel
- Kampagnen existieren nur zwei Monate
- Missbrauch melden
- Kampagnen erstellen/editieren/löschen
- Konto erstellen/editieren/löschen
- Kampagnen suchen
- Favoriten
- Bildupload
- dreisprachige Website

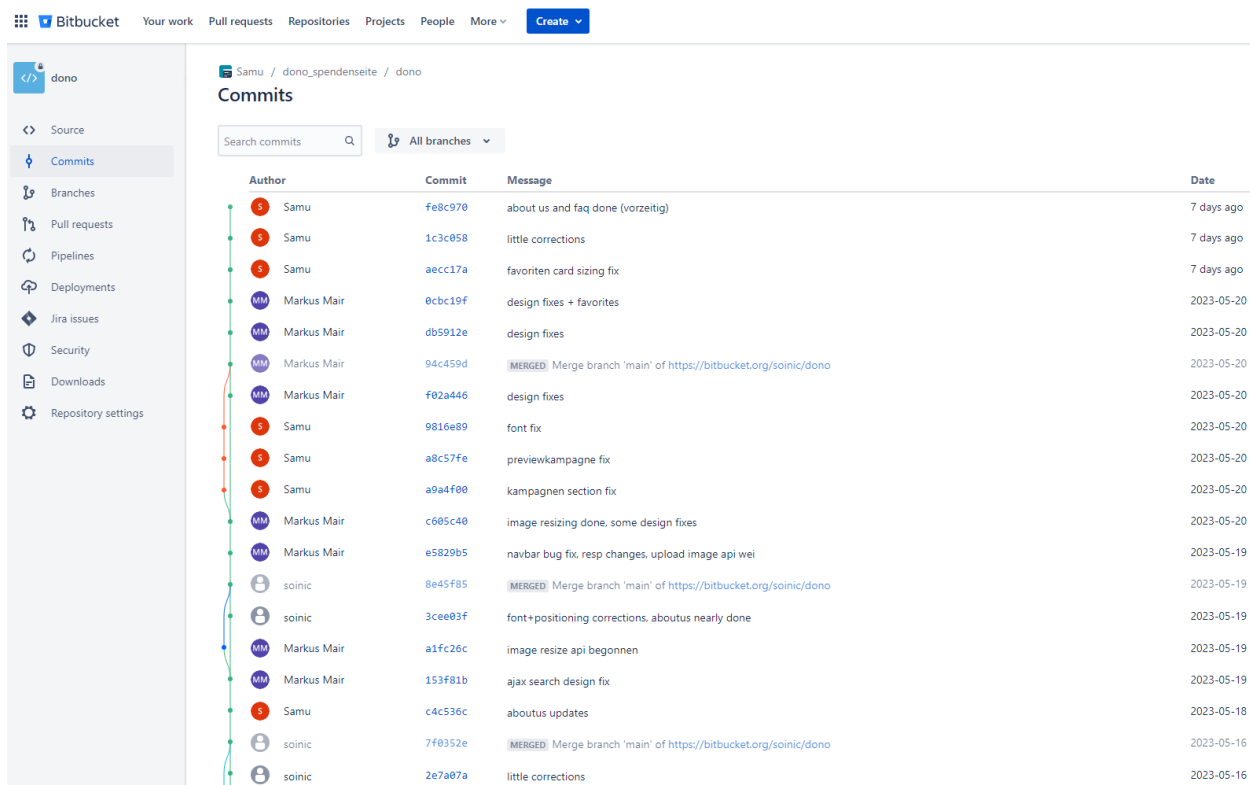
3. Ausgangssituation

3.1 Projektziel

Unser Projektziel war es eine funktionierende und gut gestaltete Website zu erschaffen, welche so gut es geht alle geplanten Funktionen abdeckt. Es soll einem Spender einfach und unkompliziert gelingen einem Spendenbetreiber Geld zu überweisen.

3.2 Projektumfeld

Gearbeitet wurde in einem der Computerräume unserer Schule. Zudem verbrachten wir auch viel Zeit zu Hause, um am Projekt zu arbeiten. Zum Synchronisieren unseres Codes verwendeten wir den webbasierten Onlinedienst „Bitbucket“ welcher sich perfekt für Software-Entwicklungsprojekte eignet.



Author	Commit	Message	Date
Samu	fe8c970	about us and faq done (vorzeitig)	7 days ago
Samu	1c3c058	little corrections	7 days ago
Samu	aecc17a	favoriten card sizing fix	7 days ago
Markus Mair	0cbc19f	design fixes + favorites	2023-05-20
Markus Mair	db5912e	design fixes	2023-05-20
Markus Mair	94c459d	MERGED Merge branch 'main' of https://bitbucket.org/soinic/dono	2023-05-20
Markus Mair	f02a446	design fixes	2023-05-20
Samu	9816e89	font fix	2023-05-20
Samu	a8c57fe	previewkampagne fix	2023-05-20
Samu	a9a4f00	kampagnen section fix	2023-05-20
Markus Mair	c605c40	image resizing done, some design fixes	2023-05-20
Markus Mair	e5829b5	navbar bug fix, resp changes, upload image api wei	2023-05-19
soinic	8e45f85	MERGED Merge branch 'main' of https://bitbucket.org/soinic/dono	2023-05-19
soinic	3cee03f	font+positioning corrections, aboutus nearly done	2023-05-19
Markus Mair	a1fc26c	image resize api begonnen	2023-05-19
Markus Mair	153f81b	ajax search design fix	2023-05-19
Samu	c4c536c	aboutus updates	2023-05-18
soinic	7f0352e	MERGED Merge branch 'main' of https://bitbucket.org/soinic/dono	2023-05-16
soinic	2e7a07a	little corrections	2023-05-16



3.3 Ist-Zustand

Anfangs hatten wir neben unserer Projektidee auch eine Webseite mithilfe der Wettbewerbsanalyse gefunden. Diese verfolgte zwar auch das gleiche Kernprinzip wie „dono“, jedoch war die Umsetzung anders. Dementsprechend konnten wir maximal das Design der Webseite in Betracht ziehen. Schlussendlich haben wir uns allerdings lediglich vereinzelt auf das Grundgerüst bezogen.

3.4 Verwendete Technologien

Zur Umsetzung unseres Projektes haben wir auf verschiedene Technologien gesetzt:

- PHP 8
- HTML 5
- Bootstrap 5
- SCSS
- JavaScript
- PHP Mailer
- SweetAlert 2

Was ist PHP?

PHP ist eine serverseitige Programmiersprache, die für die Entwicklung von Webanwendungen verwendet wird. Mit PHP können dynamische Inhalte generiert werden, wie zum Beispiel das Ausführen von Datenbankabfragen, das Erstellen von dynamischen Webseiten, das Senden von E-Mails und das Verarbeiten von Formulardaten. PHP ist kostenlos und Open-Source, was bedeutet, dass es von einer großen Gemeinschaft von Entwicklern unterstützt wird, die regelmäßig neue Funktionen und Updates bereitstellen.

Was ist HTML?

HTML steht für "Hypertext Markup Language" und ist die Standard-Markup-Sprache zur Strukturierung von Inhalten im Web. Es handelt sich um eine Auszeichnungssprache, die verwendet wird, um Texte, Bilder, Links und andere Inhalte auf einer Webseite zu definieren und zu organisieren.

Was ist SCSS?

SASS bedeutet ausgeschrieben «Syntactically Awesome Stylesheets», was so viel bedeutet wie großartiges CSS. Dementsprechend wird es genauso für das Gestalten und Positionieren von Websiteelementen verwendet.



SASS ist ein CSS-Präprozessor, was so viel bedeutet, dass es CSS zwar vereinfacht und mit vielen Funktionen erweitert, jedoch muss SCSS und SASS schlussendlich immer wieder in CSS zurückkompiliert werden, da der Browser nur CSS interpretieren und verstehen kann. Dafür braucht man einen sogenannten «SASS-Compiler», der die SASS-Sprache in normales CSS kompiliert. Solche Compiler gibt es auf allen Plattformen und Betriebssystemen; beispielsweise als Plugins für die meistens Code-Editoren oder auch, wie es professionell in einer Entwicklungsumgebung verwendet wird, als npm-package. In unserem Fall verwendeten wir ein Plugin für unsere Projektumgebung welche auf dem Namen „Visual Studio Code“ hört.

Was ist Bootstrap?

Bootstrap ist ein kostenloses und Open-Source-Framework für das Frontend-Webdesign. Es enthält vorgefertigte HTML-, CSS- und JavaScript-Komponenten und -Stile, die von Entwicklern verwendet werden können, um schnell und einfach responsive und mobile-first-Webseiten und -Anwendungen zu erstellen. Bootstrap wird von einer großen Gemeinschaft von Entwicklern und Unternehmen unterstützt. Es bietet auch eine breite Palette von Funktionen wie Grid-Systeme, Navigationselemente, Formulare, Typografie und vieles mehr.

Was ist JavaScript?

JavaScript ist eine clientseitige Programmiersprache, die in Webbrowsern ausgeführt wird. Es ermöglicht die Interaktion mit Webseiten, indem es dynamische Inhalte und Funktionalitäten bereitstellt. Mit JavaScript können Entwickler Formulare validieren, Benutzerinteraktionen wie Klicks und Hover-Events verarbeiten, Animationen erstellen und vieles mehr. Es ist auch eine der am häufigsten verwendeten Programmiersprachen der Welt und wird von einer breiten Gemeinschaft von Entwicklern und Unternehmen unterstützt.

Was ist ein PHP Mailer?

PHPMailer ist eine PHP-Bibliothek, die zur Vereinfachung des Versands von E-Mails über SMTP entwickelt wurde. Mit PHPMailer können Entwickler E-Mails über einen SMTP-Server oder lokale Mailserver senden und dabei Funktionen wie Anhänge, HTML-Formatierung und die Verwendung von CC- und BCC-Feldern nutzen. Es ist eine Open-Source-Bibliothek und wird von einer aktiven Entwicklergemeinschaft unterstützt.

PHPMailer wird häufig in Webanwendungen eingesetzt, um die Kommunikation mit Benutzern zu erleichtern.

Was ist SweetAlert?

SweetAlert ist eine Open-Source-JavaScript-Bibliothek, die zur Verbesserung der Benutzererfahrung von Meldungen und Warnungen in Webanwendungen verwendet wird. Es bietet eine einfach zu bedienende und benutzerfreundliche Alternative zu den Standard-Dialogfeldern von Webbrowsern und ermöglicht es Entwicklern, ansprechende und anpassbare Popup-Fenster mit benutzerdefiniertem Text, Schaltflächen und Animationen zu erstellen.

4. Zeitplanung und Management

4.1 Projektentwicklungsmethode

Durchgeführt wurde unser Projekt mit einer eigenen und deutlich einfacheren Art der Scrum-Methode da die ursprüngliche Variante normalerweise in größeren Gruppen verwendet wird in welcher man schnell die Übersicht verlieren kann. Wir haben uns dazu jeden Sonntag getroffen, um zu berichten was erledigt wurde bzw. noch fehlt. Gleichzeitig haben wir einen Plan erstellt in welcher die Teilarbeit, die Kategorie, der Startzeitpunkt sowie der Endzeitpunkt eingetragen werden muss.

4.2 Zeitplanung

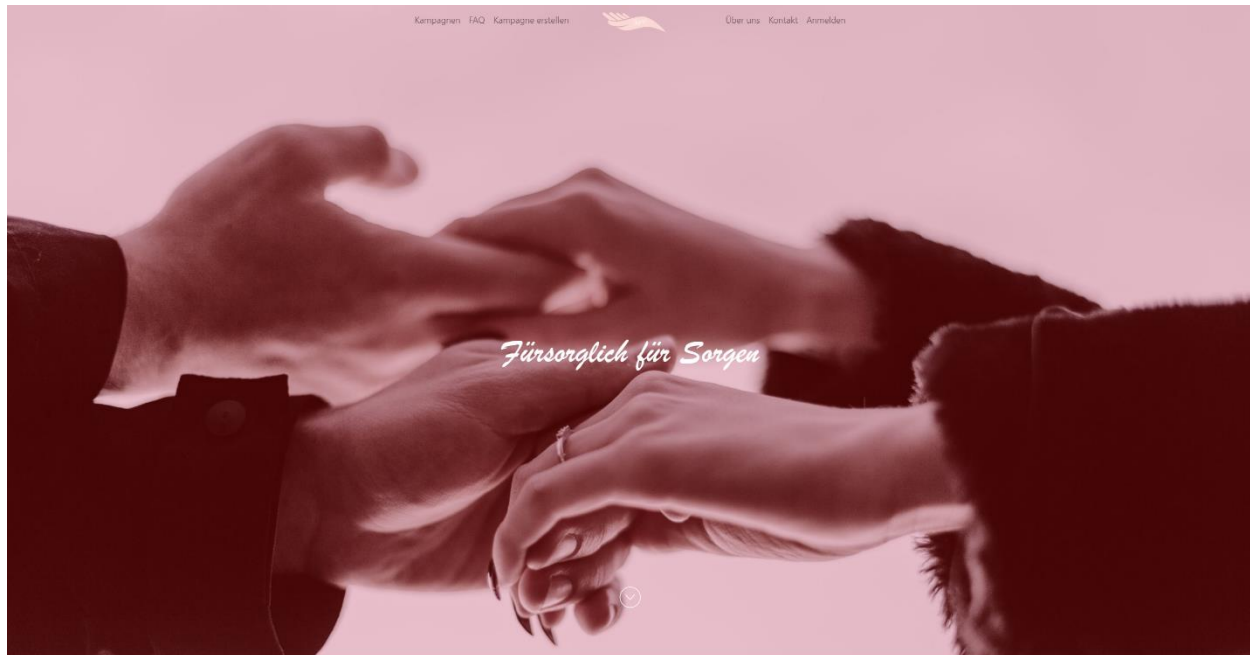
Am Anfang des Projektes haben wir nicht nur aufgeschrieben, welche Funktionen das Projekt haben soll, sondern haben auch einen Zeitplan erstellt.

Soll Zeitplan:

Kategorie	Zeit in h
Analyse	25h 30min.
Design	72h
Programmierung	104h
Datenbank	8h
Testen	6h

5. Projektumsetzung

5.1 Design/Frontend Entwicklung



Im Frontend verbrachte ich die meiste Zeit während der Umsetzung des Projektes, nachdem ich zunächst ein Wireframe (Skizzierung aller Seiten) für unsere Webseite erstellt habe. Mir war dabei besonders wichtig, dass unsere Site:

- für sämtliche Viewports (Bildschirmdiagonalen) ansprechend gestaltet ist
- übersichtlich / einfach zu bedienen ist
- zu Wohlbefinden beiträgt

Mit HTML wurde zunächst das passende Grundgerüst gebaut. Daraufhin verwendete ich Bootstrap welches besonders gute geeignet für die Erstellung der Site war, da es bereits vorgefertigte SCSS, HTML sowie JavaScript Klassen enthält. Somit konnte „dono“ langsam Gestalt annehmen. Doch Bootstrap eignet sich keineswegs für die alleinige Gestaltung der Website. Um Elemente im Detail zu verändern und anzupassen sowie für alle Bildschirmdiagonalen ansprechend zu gestalten, verwende ich zusätzlich SCSS. Es ermöglicht es HTML-Elemente anhand ihrer ID bzw. Klassennamen anzusprechen und beliebig anzupassen. Hier ein Beispiel für die Veranschaulichung:



```
1 $primary: #FBDAD6;
2 $secondary: #636363;
3 $text-color: #636363;
4
5 body {
6   position: relative;
7   min-height: 100vh;
8   display: grid;
9 }
10
11 .main_login, .main_contact, .main_faq {
12   height: 100vh;
13 }
14
15 .navbar-scrolled {
16   background-color: white !important;
17   z-index: 10 !important;
18 }
19
20 .navbar {
21   height: 3.8em;
22   position: fixed;
23   top: 0;
24   width: 100%;
25   background-color: transparent;
26   transition: background-color 0.5s ease;
27 }
28
29 .navbar-brand {
30   z-index: 5 !important;
31 }
32
33 .main-image_div {
34
35   background-image: url("../images/donatepink.jpg");
36   background-size: cover;
37   background-repeat: no-repeat;
38   height: 100vh;
39 }
40
41 .arrow-main {
42   top: 82% !important;
43 }
44 }
```

Ich greife auf HTML-Elemente zu (auf **Klassen** mit „.“, auf **IDs** mit „#“) und gestalte diese beliebig.

5.2 Backend Entwicklung

In Backend war ich lediglich bei Einzelheiten beteiligt. Dieses beschränken sich auf das Schreiben einiger Methoden zum korrekten Darstellen der Kampagnen, auf der Hauptseite sowie der Seite auf denen alle Kampagnen ersichtlich sind. Außerdem programmierte ich auch den Spamschutz des Kontaktformulars in Form eines Captchas und das Senden von Kontaktformularen an uns.

Immer noch Fragen?
Hier könnt Ihr uns kontaktieren

Vorname	Nachname
E-Mail	Titel
Beschreibung	

Ich bin kein Roboter:
bsp. 1234

Der Captcha Code lautet 5531

Abschicken

```

} else if ($fid == 8) {
    // kontakt

    if (empty(trim($_POST['contact_email'])) || empty($_POST['contact_first_name']) || empty($_POST['contact_last_name']) || empty($_POST['contact_title']) || empty($_POST['contact_description'])) {
        $errmsg = CONTACT_ERRORMSG;

        header("Location: index.php?action=contact&errmsg=$errmsg");
    } else if ((empty(trim($_POST['captcha'])))) {
        if ($_POST['captcha'] == $_SESSION['captcha']) {
            $firstname = $_POST['contact_first_name'];
            $lastname = $_POST['contact_last_name'];
            $email = $_POST['contact_email'];
            $subject = $_POST['contact_title'];
            $message = $_POST['contact_description'];

            sendContactMail($firstname, $lastname, $subject, $message, $email);

            header("Location: index.php?action=main");
        } else {
            $errmsg = "Das Captcha stimmt nicht überein";

            header("Location: index.php?action=contact&errmsg=$errmsg");
        }
    } else {
        $errmsg = "Sie müssen das Captcha ausfüllen!";

        header("Location: index.php?action=contact&errmsg=$errmsg");
    }
}

```

Zunächst wird überprüft ob auch alle erforderlichen Daten eingegeben wurden (Vorname, Nachname, E-Mail...). Wenn alle erforderlichen Daten eingegeben wurden, wird überprüft, ob der eingegebene Captcha-Code mit dem Code in der Meldung übereinstimmt.

Realisiert wurde das Backend mit PHP in seiner neusten Version (8), JavaScript und einigen Bibliotheken wie SweetAlert oder dem PHP Mailer.

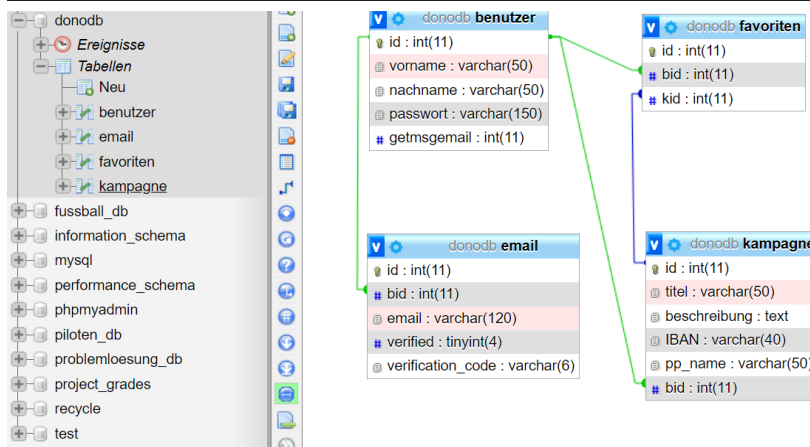
5.3 Datenbank Entwicklung

Die Datenbankentwicklung und Implementierung erledigte ich ganz am Anfang. Zunächst erstellte ich die passenden Tabellen, welche ich daraufhin miteinander verknüpfte. In ihnen werden die ganzen Benutzer sowie die jeweilige Kampagne gespeichert. Zudem können auch Kampagnen als Favoriten gespeichert werden.

```
<?php
295 references | 0 implementations
class DB
{
    /**
     * @var mixed
     */
    28 references
    private static $db = null;

    /**
     * @return mixed
     */
    263 references | 0 overrides
    public static function getDB()
    {
        if (self::$db == NULL) {
            try {
                self::$db = new PDO('mysql:host=localhost;dbname=donodb;charset=utf8', 'admin', 'dono2913');
                self::$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            } catch (PDOException $e) {
                echo $e->getMessage();
            }
        }

        return self::$db;
    }
}
?>
```



Diese Datenbank wurde schließlich mithilfe einer PHP-Funktion passend eingebunden. Wichtig war uns allerdings, dass ein Benutzer nur jeweils eine Kampagne erstellen kann. Zudem können nicht mehrere Kampagnen mit derselben IBAN oder demselben PayPal-Namen existieren. Im

Laufe des Projektes richtete ich einen sogenannten „Event Listener“ ein. Dieser bewirkt, dass Kampagnen ab dem Zeitpunkt der Erstellung nur etwa 2 Monate aktiv sind. Nach

dem Ablauf der Zeit wird die Kampagne schließlich gelöscht und der Benutzer kann eine neue erstellen.

6. Projektergebnisse

6.1 Schwierigkeiten und Hindernisse

Das größte Problem war das Finden einer alternativen Zahlungsmethode, welche ich mir zur Aufgabe gemacht habe. Dabei verging extrem viel Zeit, da die meisten Methoden (Bsp. Sofortüberweisung) ein rechtliches Hindernis darstellten. Ein großes und unumgängliches Thema, mit dem ich dauerhaft konfrontiert wurde, waren Steuerabgaben. Für eine simple Spendenseite wie „dono“ kommt dies allerdings nicht in Frage da es den Rahmen sprengen würde. Und so war ich für etwa einen Monat auf der Suche nach einem geeigneten Zahlungssystem, bei welchem eine Dokumentation zur Verfügung stand, etwa bei Klarna, Wise oder auch Stripe, was ein großer Fehler von mir war. Diese Überweisungsmethoden waren ein rechtliches Hindernis und waren nicht für eine dynamische Art und Weise ausgelegt, wie es bei unserer Site der Fall ist. Am Ende habe ich mich für eine Schlichte IBAN-Überweisungsmethode entschieden da so Nutzer einfach über ihrer Bank Geld an die IBAN des Hilfsbedürftigen spenden können. Dies ist innerhalb der EU kostenfrei, allerdings kann es möglich sein, dass die Bank eine Überweisungsgebühr bei großen Summen erhebt, welche aber einen kleinen Prozentsatz der Überweisung ausmacht. Ich habe mich dazu entschlossen die dafür aufgewendete Zeit nicht zu protokollieren, da sie dem Projekt letztendlich nicht zugutekam.

6.2 Ist-Soll Vergleich Funktionen und Zeitplanung

Nicht alle unsere geplanten Funktionen wurden letztendlich umgesetzt. Funktionen die effektiv umgesetzt wurden:

- Spenden über PayPalMe
- Alternative Spendenmethode (IBAN)
- Kampagnen existieren nur zwei Monate
- Kampagnen erstellen/editieren/löschen
- Bildupload
- Konto erstellen/editieren/löschen
- Favoriten
- Kampagnen suchen
- Missbrauch melden



- Drei-sprachige Website
- IBAN-Validierung
- Mail Funktion

Es gibt definitiv Funktionen, welche von uns nicht umgesetzt werden konnten. Diese betreffen allerdings nur den Prozess während des Spendens. Anonymes Spenden sowie Spendenziele waren für uns nicht möglich zu implementieren da die Zahlung nicht über uns abgewickelt wird bzw. es uns nicht möglich ist zu erfahren wer gespendet hat und welchen Betrag er gespendet hat. Ansonsten wurden so gut wie alle Anforderungen erfüllt.

Ist Zeitplan:

Kategorie	Zeit in h
Analyse	25h (-30min)
Design	56h (-16h)
Programmierung	79h (-25h)
Datenbank	7h 30 min (-30min)
Testen	14h (+8h)
Insgesamt: 181h 30min (-34h)	

Besonders ersichtlich ist, dass die tatsächlich benötigte Zeit deutlich unter unserer Annahme stand. Dies ist besonders in der Kategorie Design und Programmierung ersichtlich da dort auch am meiste benötigte Zeit vorausgeplant war. Doch das Testen unserer Webseite benötigte deutlich länger als zuvor geplant. Dies hat damit zu tun, dass im Laufe des Projektes immer wieder Zwischentests stattgefunden haben.

6.3 Fazit

Im Großen und Ganzen bin ich mit unserer Leistung zufrieden, jedoch hätte ich mir dennoch gewünscht, dass die Funktion mit der Stripe API funktioniert hätte. Stattdessen habe ich dort viel Zeit liegen lassen. Das Projekt in dieser Konstellation ist uns gut geglückt und jeder trug seinen Beitrag zum Erstellen des Projektes bei. Es war eine äußerst gute Möglichkeit sich weiterzubilden und an seinen Schwächen zu arbeiten bzw. etwas Neues dazulernen. Es gab öfters Momente, wo wir an der Schwelle standen, doch mit viel Mühe und Überwindung konnten wir Probleme hinter uns lassen.

7. Selbständigkeitserklärung

Hiermit erkläre ich Samuel Soini, dass ich die Dokumentation selbst verfasst habe. Ich bin mir bewusst, dass falsche Angaben und Plagiate schwerwiegende Konsequenzen für mich haben können.

8. Quellen

Was ist PHP?

- <https://www.php.net/manual/de/intro-whatism.php>

Was ist HTML?

- https://de.wikipedia.org/wiki/Hypertext_Markup_Language
- <https://www.seo-kueche.de/lexikon/html/>

Was ist SCSS?

- <https://nextgeneration.mysign.ch/de/themen/wissen/frontend-entwicklung/scss-sass.html>

Was ist Bootstrap?

- <https://it-talents.de/it-wissen/bootstrap/>

Was ist JavaScript?

- <https://www.seobility.net/de/wiki/JavaScript>

Was ist PHP Mailer?

- <https://en.wikipedia.org/wiki/PHPMailer>
- <https://www.ionos.de/digitalguide/e-mail/e-mail-technik/phpmailer/>

Was ist SweetAlert?

- <https://www.drweb.de/sweetalertjs-schickere-flexiblere-alternative-zu-alert-52247/>